# CRIPTOGRAFÍA BASADA EN GRUPOS

**Trabajo de fin de Máster**

*Máster en Modelización e Investigación Matemática, Estadística y Computación*

Facultad de Ciencias

Universidad de Zaragoza

**Laura Miralles Millas**

Directora del trabajo: Conchita Martínez Pérez

4 de diciembre de 2020

*Ten en cuenta que la información es poder*

Bill Gates

A mis padres.
A mi hermana.

# Resumen

La *criptografía* se define como el método de proteger la información y la comunicación a través de códigos capaces de leer y procesar esa información. Son muchos los métodos criptográficos desarrollados a lo largo de la historia pero casi todos con un trasfondo común, la resolución de un problema matemático. Muchos de los ellos se han estudiado en el campo de la teoría de número, sin embargo, en las últimas décadas se han empezado a estudiar criptosistemas basados en la teoría combinatoria de grupos. La motivación de los autores que han investigado criptosistemas de clave pública es conseguir una alternativa segura basada en la dificultad de resolver problemas de búsqueda dentro de la teoría combinatoria de grupos, como son: el problema de la palabra y el problema de la conjugación.

El objetivo de este trabajo es estudiar diferentes esquemas criptográficos usando estos problemas en los grupos de trenzas. Para ello, en primer lugar, estudiaremos la definición y las propiedades de estos grupos de trenzas, denominados así por su interpretación geométrica. Estos grupos son muy interesantes, en particular porque existen diversas formas normales para sus elementos. A lo largo del capítulo 2, nos centraremos en probar una de esas formas canónicas que permite reescribir las palabras que representan a las trenzas en una expresión única. La idea es poder usarla en la generación de claves públicas es los esquemas criptográficos. El resultado es que parte de la palabra original queda oculta y dificulta que un adversario ataque el esquema.

En el tercer capítulo, comenzaremos la sección con el protocolo de intercambio de claves propuesto por Diffie y Hellman. A continuación, se considerarán dos esquemas basados en el problema de la palabra (criptosistema de Shpilrain y Zapata, y la construcción de Wagner y Magyarik) que pueden ser aplicados a grupos de trenzas y otros dos esquemas (el protocolo de Ko-Lee y el protocolo de Anshel, Anshel y Goldfeld) que usan grupos de trenzas en el problema de la conjugación. Finalmente, se describirá la firma digital WalnutDSA. Esta firma se caracteriza por usar dos trenzas como clave secreta y un mensaje encriptado. Para ello, aplicaremos los resultados del segundo capítulo así como dos nuevos conceptos. A lo largo largo del trabajo, se ilustran varios ejemplos con la finalidad de facilitar la comprensión de los razonamientos y resultados.

**Palabras clave:** Grupo de trenzas, trenzas positivas, representación coloreada de Burau, criptografía, protocolo de Diffie-Hellman, problema de la palabra, problema de la conjugación, protocolo de Anshel-Anshel-Goldfeld, protocolo de Ko-Lee, algoritmo de Wagenr y Magyarik, WalnutDSA.

# Abstract

*Cryptography* is a method of protecting information and communication through the use of codes, so that only those for whom the information is intended can read and process it. Many cryptosystems have emerged throughout history, all with a common background: solving a mathematical problem. Although many of them are based on the number theory, in the last decades, cryptosystems based on the combinatorial group theory have begun to be studied. In search for more efficient and secure alternatives to establish cryptograpic protocols, several authors have come up with public key establishment protocols as well as with public key cryptosystems based on hard search problems from combinatorial group theory such as: the word problem and the conjugacy problem.

The goal of this work is to study some different cryptographic schemes by using these problems in the braid groups. So, firstly, we are going to study the definition and properties of braid groups, named like this because of its geometric interpretation. These groups are really interesting, in particular, there are several different normal form for elements of a braid group. Throughout chapter two, we will focus on clearly proving one of these canonical forms to rewrite words representing braids as an unique expression. This expression can be used to generate the public keys in the cryptograpic schemes. The result is that part of the original word is hidden, so it is more difficult attack the scheme.

In the third chapter, we will begin the section about the Diffie-Hellman key exchange. After that, we are going to consider one scheme based on the word problem (Shpilrain-Zapata cryptosystem) and applied to braid groups, and two protocols (Ko-Lee protocol and Anshel-Anshel-Goldfeld protocol) by using braid groups. Finally, we are going to describe the digital signature WalnutDSA. This signature needs two secret braids and the encrypted message. So, we will apply the results of the second chapter as well as two new concepts. Throughout the work, many examples will be presented in order to clarify and justify some results and observations.

**Key words:** Braid group, positive braids, Colored Burau Representation of the Braid Group, cryptography, Diffie-Hellman key exchange, Word problem, Congujacy problem, Anshel-Anshel-Goldfeld protocol, Ko-Lee protocol, Wagner-Magyarik cryptosystem, WalnutDSA.

# Índice general

**Resumen** v

**Abstract** vii

**1. Preliminares** 1
1.1. Introducción . . . 1
1.2. Conceptos de teoría combinatoria de grupos . . . 2
1.3. Conceptos previos de criptografía . . . 7
1.4. Conceptos previos de teoría computacional . . . 8

**2. Grupos de trenzas** 11
2.1. Definición de grupo de trenzas . . . 11
2.2. Trenzas positivas . . . 14
2.3. Forma normal ponderada a la izquierda . . . 18
2.4. Representación coloreada de Burau . . . 23

**3. Problemas algorítmicos en grupos de trenzas** 27
3.1. Protocolo de Diffie-Hellman . . . 27
3.2. Problema de la palabra . . . 29
3.2.1. Esquema de Shpilrain-Zapata . . . 30
3.2.2. Algoritmo de Wagner y Magyarik . . . 34
3.3. Problema de la conjugación . . . 35
3.3.1. Protocolo de Ko-Lee . . . 37
3.3.2. Protocolo de Anshel-Anshel-Goldfeld . . . 41

**4. Algoritmo WalnutDSA** 45
4.1. E-multiplicación . . . 45
4.2. Elementos de "camuflaje" . . . 46
4.3. Descripción del algoritmo . . . 48

**5. Conclusiones** 53
5.1. Futuras líneas de trabajo . . . 53
5.2. Conclusión . . . 54

**Bibliografía** 57

# Capítulo 1

# Preliminares

## 1.1. Introducción

La *criptografía* es la ciencia que estudia la protección de la información y de la comunicación a través de diferentes algoritmos. La historia de la criptografía se remonta al siglo V a.C. donde ya se usaban métodos como el de trasposición, basado en enrollar el mensaje sobre un cilindro, o el de sustitución, donde cada letra se asocia a un número. Desde entonces han surgido multitud de algoritmos, algunos de cifrado simétrico (lo cual significa que el método que se usa para desencriptar es el proceso inverso al de encriptar) y otros de cifrado asimétrico (es decir, el método de encriptar y desencriptar no es equivalente).

En el primer capítulo se van a revisar algunas nociones básicas de teoría combinatoria de grupos (Sury [27], Myasnikov [21], [9]), que nos servirán para comprender todo el desarrollo matemático que llevan implícitos los protocolos que se desarrollan en los capítulos 2 y 3, así como conceptos de criptográfica (Koblitz [18]) que nos ayudarán a interpretar mejor los algoritmos criptográficos.

Los protocolos que se describen a lo largo del trabajo se fundamentan en resultados de teoría de grupos y en ellos, la estructura subyacente del grupo juega un papel importante. A estos grupos se les denomina grupos plataforma. En este trabajo se considerarán como grupos plataforma a los grupos de trenzas. Por este motivo, introduciremos en el capítulo 2 la definición del grupo de trenzas $B_N$ (Kassel [17]), denominado así porque geométricamente se interpreta como un conjunto de $N$ cuerdas que parten de una posición $i$ de una barra superior, entrelazándose (o no) con el resto de cuerdas hasta llegar a una posición $j$ de una barra inferior para $i, j \in \{1, \ldots, N\}$. El diagrama que representan induce una permutación del conjunto $\{1, \ldots, N\}$, lo cual incita a definir las *trenzas puras* (trenzas cuya permutación inducida es la identidad). Además, estas trenzas pueden escribirse de manera única mediante lo que se llama una forma canónica, para las trenzas existen varias: la de Birman (Birman [4]), la de Garside (Garside [10]), la forma normal a izquierda (H.H.Ko [14]), etc. Se definirá esta última, que en definitiva es una versión mejorada de la forma canónica de Garside, y será de gran utilidad para reescribir la palabra que represente a una trenza dada en otra palabra equivalente. Al final del capítulo 2, se explica la representación coloreada de Burau (S.J. Lee [15], Kotov [19]) asociada a $B_N$, de modo que a cada trenza se le asocia un tupla $(M, \pi)$ consistente en una matriz $M$ y una permutación $\pi$. Este concepto será de gran interés tanto en el protocolo de Anshel-Anshel-Goldfeld, que lo utiliza en su extractor de claves (función que a un elemento le asocia una clave) como en la firma digital de WalnutDSA.

En 1976 nació la noción de criptografía de clave pública con Diffie y Hellman (Shpilrain [23], Glez [11]), aunque el primer y más conocido algoritmo de este tipo y válido tanto para cifrar como para firmar digitalmente es el algortimo RSA. Sin embargo, en este trabajo nos centraremos en criptosistemas asimétricos. El capítulo 3 comienza con el estudio previo del intercambio de claves propuesto por Diffie y Hellman. A continuación, se presentarán dos importantes problemas motivados por algoritmos criptográficos, cuya dificultad ha garantizado en muchos casos la seguridad del esquema en cuestión: problema de la palabra (Wagner [29], Miasnikov [21], Birman [4], Shpilrain [25]) y el problema de la conjugación (Shpilrain [26], H.H. Ko [14], Anshel [2]). Para el primero de ellos, se presentarán dos sistemas criptográficos: El protocolo de Shpilrain-Zapata (proponen un criptosistema basado en las presentaciones de un grupo elegido [25]) y la construcción de Wagner y Magyarik (plantea la idea de que dado un grupo donde no se puede resolver el problema de la palabra, se pase a un grupo cociente donde sí se pueda [29]). Por otro lado, se basan en el problema de la conjugación: el protocolo de Ko-Lee (propone un intercambio de claves y un criptosistema basado en la elección de dos subgrupos que conmuntan [14]) y el protocolo de Anshel-Anshel-Goldfeld (describe un intercambio de claves en el que la clave compartida se obtiene de la representación de Burau [2]).

Finalmente, en el capítulo 4 se describirá la firma digital WalnutDSA, una firma que puede ser escrita como una palabra $W_1 \cdot E \cdot W_2$ donde $W_1, W_2$ son dos trenzas secretas y $E$ es el mensaje encriptado. El proceso que conlleva la generación de la firma requiere de dos nuevos conceptos: unos elementos estabilizadores denominados elementos de "camuflaje" (Kotov [19], Blackburn [5]) y la función de una-vía denominada E-multiplicación. Esta última tiene un gran interés, ya que ha sido considerada de alta resistencia cuántica, una muy buena propiedad de cara al futuro y al avance de la criptografía cuántica, que es el campo que persigue encontrar un método eficiente para intercambiar claves utilizando las leyes de la física cuántica, por medio de la transmisión de fotones polarizados al azar en distintas direcciones.

## 1.2. Conceptos de teoría combinatoria de grupos

La teoría de grupos combinatoria se basa en utilizar presentaciones de grupos para verlos como objetos combinatorios. En esta teoría es esencial el uso de diferentes técnicas vinculadas a varias ramas de las matemáticas, como la topología, lógica, teoría de números, teoría de grupos, etc. Los términos iniciales relativos a álgebra ayudarán a comprender el concepto de grupo de trenzas en el siguiente capítulo. Entre los conceptos más importantes destacan la noción de grupo, grupo libre y presentación de un grupo.

**Definición.** Un *grupo* es un conjunto $G$ dotado de una operación binaria interna '$\cdot$' que verifica las siguientes propiedades:

- '$\cdot$' es asociativa, es decir, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$, para todo $a, b, c \in G$.
- '$\cdot$' posee un elemento neutro, denotado $e$. Este elemento cumple: $e \cdot a = a \cdot e$, para todo $a \in G$.
- Cada elemento $a \in G$ posee un elemento simétrico, es decir, existe $a^{-1} \in G$ tal que $a \cdot a^{-1} = e = a^{-1} a$.

Si $H$ es un subconjunto de $G$ y también verifica las propiedades anteriores, entonces se dice que $H$ es un *subgrupo* de $G$, y se denota $H \leq G$.

**Definición.** Sea $G$ un grupo y $N \leq G$. Se dice que $N$ es un grupo *normal*, y se denota $N \trianglelefteq G$, si es invariante por conjugación, es decir, dado un elemento $x \in N$ y $g \in G$, el elemento $\tilde{x} = gxg^{-1} \in N$. Se dice que $\tilde{x}$ es el *elemento conjugado* de $x$ por el elemento $g$, y se denota $\tilde{x} \sim x$.

Consideremos a partir de ahora un conjunto arbitrario $X$.

**Definición.** Sea $w$ un elemento de $X$. Se dice que $w$ es una *palabra* si es una cadena finita de elementos (posiblemente repetidos) que podemos escribir de la siguiente forma

$$w = y_1 \cdots y_n \quad \text{donde} \quad y_i \in X \tag{1.1}$$

Al número $n$ de elementos que forman la secuencia de $w$ lo denominaremos *longitud* de la palabra, y lo denotaremos como $l(w)$.
La *palabra vacía* la representaremos con la letra $\varepsilon$ y se considera que es una palabra de longitud cero, $l(\varepsilon) = 0$.

Consideramos el conjunto

$$X^{-1} = \{x^{-1} | x \in X\}$$

donde realmente $x^{-1}$ es una expresión formal compuesta por los símbolos **x**, y $-\mathbf{1}$ (es decir, no representa literalmente el elemento inverso). Denotamos entonces

$$X^* = X \cup X^{-1}$$

de modo que para cada $y \in X^*$ el elemento inverso $y^{-1}$ es

$$y^{-1} = \begin{cases} x^{-1} & \text{si } y = x \in X \\ x & \text{si } y = x^{-1} \in X^{-1} \end{cases}$$

Así, una expresión como (1.1) pero sobre $X^*$ se define como

$$w = x_{i_1}^{\varepsilon_1} \cdots x_{i_n}^{\varepsilon_n} \quad \text{siendo } x_{i_j} \in X,\ \varepsilon_j \in \{1, -1\}.$$

Al conjunto de palabras que se obtienen de esta forma lo denotamos $W(X^*)$, es decir,

$$W(X^*) = \{w = y_1 \cdots y_n \mid y_i \in X^*\}.$$

**Definición.** Sea una palabra $w \in W(X^*)$, si $w$ tiene una subpalabra de la forma forma $yy^{-1}$ o de la forma $y^{-1}y$ diremos que se puede realizar una cancelación. Y en otro caso, diremos que $w$ es *reducida*. Es decir, una palabra es reducida si:

- Es una palabra vacía.
- Es de la forma $w = y_1 \cdots y_n$ y no contiene una subpalabra de la forma $yy^{-1}$ para $y \in X^*$.

Para obtener una palabra reducida se aplicará un proceso de reducción $\rho : X^* \to X^*$ que comience en $w$ y consista en eliminar de forma progresiva las subpalabras de la forma $yy^{-1}$ con $y \in X^*$ hasta terminar en una palabra reducida $w_n$:

$$w \to w_1 \to \cdots \to w_n.$$

Los pasos a seguir en este proceso de reducción no están determinados de forma única, sin embargo, el siguiente lema demuestra que la palabra $w_n$ final será la misma.

**Lema 1.1** (Ver [27], proposición 1, p.4)**.** *Sea una palabra $w \in X$. Dadas dos reducciones*

$$w \to w_1' \to \cdots \to w_n'$$

$$w \to w_1'' \to \cdots \to w_n''$$

*la palabra reducida es la misma, es decir, $w_n' = w_n''$.*

**Proposición 1.2.** *Sea $w$ una palabra. Existe una única forma reducida $w_n$ para $w$.*

Este último resultado motiva una relación de equivalencia, de manera que si dos palabras tienen la misma forma reducida, entonces si dirá que son equivalentes.

Suponer ahora que tenemos un grupo $G$. Si $X \subseteq G$ entonces una palabra de la forma $w = x_{i_1}^{\varepsilon_1} \cdots x_{i_n}^{\varepsilon_n}$ determina un único elemento en $G$ que es igual al producto $x_{i_1}^{\varepsilon_1} \cdots x_{i_n}^{\varepsilon_n}$ con cada elemento $x_{i_j}^{\varepsilon_j} \in G$.

**Definición.** Un grupo $G$ se dice *grupo libre* si existe un subconjunto $X$ de $G$ tal que todo elemento de $G$ puede escribirse de manera única como una palabra reducida en los elementos de $X$ y de sus inversos. En este caso se dice $X$ es una base de $G$.

Equivalentemente, el grupo $G$ se dice libre si toda palabra reducida en $X$ define un elemento no trivial (distinto de la identidad) de $G$. Al grupo libre generado por $X$ se denota $F(X)$. En este caso se puede decir también que $G$ es libre en $X$ o que $G$ está generado por $X$, por lo que $F(X) = G$. Si $X$ es finito, se dice que $G$ está finitamente generado.

En el conjunto de clases de equivalencia de $W(X^*)$ respecto a la relación anterior se puede definir un producto mediante yuxtaposición y probar que se obtiene un grupo libre de base X, por lo tanto tenemos:

**Teorema 1.3.** *Dado un conjunto $X$ siempre existe un grupo libre $F$ con base $X$.*

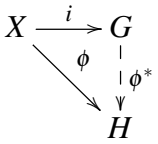
Presentamos a continuación una de las caracterizaciones más importantes de los grupos libres, para lo cual necesitamos la siguiente definición.

**Definición.** Un subconjunto $X$ de un grupo $G$ es un *conjunto de generadores* si cada elemento $g \in G$ se puede expresar como un producto finito de los elementos de $X$ y sus inversos. Es decir,

$$g = x_1^{n_1} \cdot x_2^{n_2} \cdots x_k^{n_k}, \quad n_i \in \mathbb{Z}.$$

Si $X$ es finito, se dirá que $G$ está finitamente generado.

**Teorema 1.4.** *Sea $X$ un conjunto de generadores de un grupo $G$, $X \subseteq G$. El grupo $G$ es un grupo libre con base $X$ si y solo sí se verifica la siguiente propiedad universal: toda función $\phi : X \to H$ de $X$ en un grupo $H$ se puede extender a un único homomorfismo $\phi^* : G \to H$, de modo que el siguiente diagrama conmuta*




*(Observar que $i : X \to G$ es la inclusión de $X$ en $G$).*

*Demostración.* ($\Rightarrow$) Sea $G$ un grupo libre en $X$ y sea $\phi : X \to H$. Como $G$ es libre, cada $g \in G$ está definido por una única palabra reducida $w \in W(X^*)$, es decir

$$g = x_{i_1}^{\varepsilon_1} \cdots x_{i_n}^{\varepsilon_n} \quad \text{con} \quad x_{i_j} \in X,\ \varepsilon_j \in \{1,-1\}.$$

Definimos

$$\phi^* = \phi(x_{i_1})^{\varepsilon_1} \cdots \phi(x_{i_n})^{\varepsilon_n}. \tag{1.2}$$

Veamos si $\phi^*$ es un homomorfismo. Sean $g,h \in G$, podemos escribir en la siguiente forma sus correspondientes palabras reducidas en $X^*$, siendo $y_i, z_j \in X^*$ tales que $y_n y_{n+1} \neq 1$,

$$g = y_1 \cdots y_n \cdot z_1 \cdots z_m$$

$$h = z_m^{-1} \cdots z_1^{-1} \cdot y_{n+1} \cdots y_k.$$

Entonces,

$$gh = y_1 \cdots y_n \cdot y_{n+1} \cdots y_k$$

sigue siendo una palabra reducida y no vacía que representa al elemento $gh$. Es más,

$$\begin{aligned}\phi^*(gh) =& \phi^*(y_1) \cdots \phi^*(y_n) \cdot \phi^*(y_{n+1}) \cdots \phi^*(y_k) \\ =& \phi^*(y_1) \cdots \phi^*(y_n)\phi^*(z_1) \cdots \phi^*(z_m)\phi^*(z_m^{-1}) \cdots \phi^*(z_1^{-1}) \cdot \phi^*(y_{n+1}) \cdots \phi^*(y_k) \\ =& \phi^*(g)\phi^*(h).\end{aligned}$$

En efecto, $\phi^*$ es un homomorfismo y además, extiende a $\phi$ ( ya que $\phi^*(g) = \phi(x)$ cuando $g = x \in X$, ver (1.2)).
Suponer que existiese otro homomorfismo $\varphi : G \to H$ verificando las hipótesis, entonces $\varphi \circ i = \phi = \phi^* \circ i$. Teniendo en cuenta que ambos son homomorfismos de grupos se deduce que $\varphi = \phi^*$. Por tanto, $\phi^*$ es única salvo isomorfismos y así $G$ verifica la propiedad universal.

($\Leftarrow$) Suponer que $G$ es un grupo que verifica la propiedad universal, y que tiene $X$ como conjunto de generadores. Considerar $H = F(X)$ (es posible por 1.3) y $\phi : X \to F(X)$ definida como $\phi(x) = x$. Entonces por hipótesis podemos extender $\phi$ a un único homomorfismo $\bar{\phi} : G \to F(X)$.
Sea $w$ una palabra reducida y no vacía en $X$, entonces hay un elemento $g \in G$ tal que $\bar{\phi}(g) = w \in F(X)$. Luego, $\bar{\phi}(g) \neq 1$. Por tanto, $g \neq 1$. Así hemos demostrado que cada palabra reducida y no vacía de $X$ define un elemento no trivial, lo que equivale a decir que todos elementos de $G$ se escriben de manera única con elementos de $X$.

□

Esta propiedad nos será de utilidad para introducir a continuación un concepto importante: la presentación de un grupo, en el cual se basa la descripción de los grupos de trenzas en el capítulo 2.

**Teorema 1.5** (Ver [27], Teorema de Nielsen-Schreier, p.32)**.** *Sea $G$ un grupo libre y $H \leq G$ un subgrupo de $G$, entonces $H$ es libre.*

## Presentación de un grupo

La propiedad universal de los grupos libres nos permite describir dichos grupos mediante generadores y relaciones (o relatores).

Por el teorema 1.3 sabemos que para todo conjunto $X$ existe el grupo libre $F(X)$, luego por la propiedad universal (ver 1.4) existe un único homomorfismo $\Phi : F(X) \to G$ tal que $\Phi(x) = x$ para todo $x \in X$. De ello se deduce que $\Phi$ es suprayectiva, luego por el primer teorema de isomorfía [1]

$$G \simeq F(X)/ker(\Phi).$$

De lo anterior se deduce la siguiente consecuencia directa:

**Proposición 1.6.** *Cada grupo G es isomorfo al cociente de un grupo libre.*

Entonces, a partir de $F(X)$ se construyen todas las palabras de $G$, y $\ker(\Phi)$ es el conjunto de palabras triviales de $G$. A cada $w \in \ker(\Phi)$ se denomina *relación (o relator)*.

Dado un subconjunto $R \subseteq \ker(\Phi)$ que genera $\ker(\Phi)$ como subgrupo normal de $F(X)$ (es decir, $R$ es el menor subgrupo normal de $F(X)$ que contiene a $R$) entonces se dice que $R$ es un *conjunto de relaciones*.

**Definición.** Se dice *presentación* del grupo $G$ al par $\langle X \mid R \rangle$ que determina $G$, salvo isomorfismo, donde $X$ es el conjunto de generadores y $R$ el conjunto de relaciones. Es decir, si denotamos por $\langle\langle R \rangle\rangle$ al menor subgrupo normal de $F(X)$ que contiene a $R$,

$$G \cong \langle X \mid R \rangle \Leftrightarrow G \simeq F(X)/\langle\langle R \rangle\rangle.$$

Diremos que es *finitamente presentado* si los conjuntos $X$ y $R$ son finitos.

$$G \cong \langle x_1, \ldots, x_n \mid r_1, \ldots, r_k \rangle.$$

**Ejemplo 1.** Se muestran a continuación algunos ejemplos de presentaciones de grupos

- Los grupos cíclicos finitos de orden $n$, $\mathbb{Z}_n : \langle g \mid g^n \rangle$.
- La suma directa $\mathbb{Z} \oplus \mathbb{Z} : \langle x, y \mid [x, y] \rangle$.
- El grupo simétrico $S_3$: $\langle x, y \mid x^3, y^2, yxyx \rangle$.

**Proposición 1.7** (Ver [27], (ii)-Lema, p.22)**.** *Todo grupo finito es finitamente presentado.*

**Definición.** Una presentación de $G$ se dice *recursiva* si los conjuntos $X$ y $R$ son enumerables de forma recursiva.

Cada grupo presentado de forma finita se presenta de forma recursiva, pero hay grupos presentados de forma recursiva que no se pueden presentar de forma finita.

Se concluye la sección definiendo el rango de un grupo libre. Suponer que $X_1, X_2$ son conjuntos finitos, entonces:

**Teorema 1.8** (Ver [9], sección 1.5 )**.** *Si* $F(X_1) \cong F(X_2)$ *entonces* $|X_1| = |X_2|$.

[1] **Primer teorema de isomorfía**. Sea $f : G \to H$ un homomorfismo de grupos. Entonces existe un isomorfismo $\bar{f} : G/(\ker\ f) \to Im\ f$

**Corolario 1.9** (Ver [9], sección 1.5)**.** *Si* $|X_1| = |X_2| \Leftrightarrow F(X_1) \cong F(X_2)$.

Estos dos últimos resultados nos muestran la estrecha relación entre un grupo libre $F$ y la cardinalidad de la base de $F$, por lo que podemos definir el *rango* de un grupo libre como la cardinalidad del conjunto $X$ sobre el que es libre. Luego el rango de $F$ es un invariante que caracteriza a $F$ y dependerá de la elección del conjunto $X$.

Un ejemplo son los grupos cíclicos infinitos, que son libres de rango 1.

## 1.3. Conceptos previos de criptografía

La criptografía de clave pública es un área bastante reciente que surgió en 1976 y desde entonces sigue muy activa. En este apartado se introducen algunas nociones necesarias para comprender el contenido criptográfico. Un *protocolo* es un algoritmo compuesto de muchas partes, que se define mediante una secuencia de pasos, en los que se especifican las acciones que requieren dos o más partes para conseguir un objetivo. Por lo general, se tratará de dos usuarios, Alice y Bob, que quieren comunicarse de forma confidencial y para ello tendrán un "protocolo" a seguir. En los capítulos 3 se exponen algunos intercambios de claves y criptosistemas, y en el capítulo 4 una firma digital.

**Definición.** Un *intercambio de claves* es un método mediante el cual dos o más personas se ponen de acuerdo en un valor para la información secreta compartida. A la información secreta compartida se le llama *clave* y se usa en un algoritmo criptográfico.

Algunos protocolos de intercambio de claves son el propuesto por Diffie-Hellman (sección 3.1), el de Ko-Lee (sección 3.3.1) o el de Anshel-Anshel-Goldfeld (sección 3.3.2).

Por otro lado, los criptosistemas de clave pública son esquemas que no requieren, en principio, un previo intercambio de claves entre los dos usuarios que se vayan a comunicar, aunque muchos de ellos se han basado en protocolos de intercambio de claves. Más formalmente,

**Definición.** Un esquema de encriptación o criptosistema es una tupla $(P,C,K,E,D)$ que verifica:

- $P$ es el conjunto de mensajes sin cifrar que se quieren enviar (en inglés, plaintext space).
- $C$ es el conjunto de mensajes cifrados (en inglés, ciphertext space).
- $K$ es el conjunto de claves públicas y privadas.
- $E = \{E_k : k \in K\}$ es una familia de funciones $E_k : P \to C$ que se denominan funciones de encriptación.
- $D = \{D_k : k \in K\}$ es una familia de funciones $D_k : C \to P$ que se denominan funciones de desencriptación.

Además, se pide que para cada $e \in E$, exista un elemento $d \in K$ tal que $D_d(E_e(p)) = p$ para todo $p \in P$.
En general, si tenemos dos usuarios que se quieren comunicar, Alice y Bob, entonces Alice puede usar un criptosistema para comunicar un mensaje secreto $m$ a Bob. Para ello usará una

clave de encriptación $e$ y generará $c = E_e(m)$ y lo mandará a Bob, que usando la clave de desencriptación $d$ obtendrá el mensaje de Alice $D_d(c) = D_d(E_e(m)) = m$.

Se presentarán algunos criptosistemas como ElGamal (sección 3.1), el esquema de Shpilrain-Zapata (sección 3.2.1) o el de Ko-Lee (sección 3.3.1).

En 1976, Diffie-Hellman introdujeron la noción de *firma digital*, aunque fue el algoritmo RSA el primero que permitió construirlas. La firma digital es un mecanismo de autenticación para que la persona que recibe el mensaje pueda verificar que efectivamente el mensaje recibido es el que le mandó la persona de origen, es decir que ningún adversario lo ha atacado o modificado. Formalmente definimos,

**Definición.** Un *esquema de firma digital* es una tupla $(K,S,V)$ que verifica:

- $K$ es el conjunto que genera la clave pública y la clave privada.
- $S$ es el conjunto de firmas digitales, que se crean a partir del mensaje que se quiere enviar, la clave privada y la clave pública.
- $V$ es la respuesta del algoritmo afirmando o negando que la firma es válida.

Algunos de ellos requerirán de una función previa que se caracteriza, entre otras propiedades, por ser de bajo costo y uniforme.

**Definición.** Una función *hash* $H : U \to \{0,1\}^n$ es una función computable mediante un algoritmo, que tiene de entrada un conjunto $U$ de elementos y los convierte en un conjunto de bits de longitud $n$.

## 1.4. Conceptos previos de teoría computacional

En el capítulo 3 se describen algunos protocolos criptográficos que han destacado por basar su seguridad en la complejidad computacional de algunos problemas matemáticos. Una de las características importantes al construir un algoritmo es el uso de una función que es fácil de computar en un sentido, pero difícil en el sentido inverso si no se tiene información adicional. A esta función se le denomina *función de una vía*.

**Ejemplo 2.** Algunas funciones de una vía que se consideran son:

$$\text{(exponencial)}\ \ f : G \longrightarrow G \qquad\qquad \text{(conjugación)}\ \ f : G \longrightarrow G$$
$$g \longmapsto g^a \ \ (a \in \mathbb{Z}) \qquad\qquad g \longmapsto aga^{-1} \ \ (a \in G)$$

En 1965, Edmonds definió un "buen" algoritmo como uno que tuviese tiempo de ejecución acotado por un polinomio, es decir, con un tiempo de ejecución polinómico. Esta característica también se tendrá en cuenta en los criptosistemas que se describen, ya que si el problema no se puede resolver en tiempo polinómico, se considerará de mayor dificultad. A continuación, se define formalmente un algoritmo polinomial. No obstante, el presente trabajo estudiará los diferentes protocolos sin profundizar en el estudio computacional que asegura la eficacia de cada uno de ellos ni en los posibles ataques que un adversario podría realizar.

**Definición** (Tiempo de ejecución)**.** El tiempo de ejecución de un algoritmo es el máximo número de operaciones elementales (operaciones bit a bit) que realiza al ejecutarse a partir de un cierto valor de entrada.

**Definición.** Sean $f$ y $g$ funciones definidas de $\mathbb{N} \to \mathbb{R}$ que toman valores siempre positivos a partir de cierto número natural. Se dice que

- $f = O(g)$ si existe una constante positiva $c$ y un natural $n_0$ tal que $\forall n \geq n_0$.

$$0 \leq f(n) \leq cg(n)$$

- $f = w(g)$ si existe una constante positiva $c$ y un natural $n_0$ tal que $\forall n \geq n_0$.

$$0 \leq cg(n) \leq f(n)$$

- $f = o(g)$ si para toda constante positiva $c$ existe un natural $n_0$ tal que $\forall n \geq n_0$.

$$0 \leq f(n) \leq cg(n)$$

**Definición** (Tiempo polinomial)**.** Un algoritmo se dice polinomial si su tiempo de ejecución es una función $O(g)$, siendo $g(x) = x^k$, con $k$ un número natural cualquiera. Todo algoritmo que no cumpla esta condición se dirá exponencial.

En el capítulo 3 veremos algunos de los problemas más famosos en teoría de grupos computacional: el problema de la palabra y el de la conjugación. En algunos grupos no existe ningún algoritmo para resolverlos, en cambio en otros hay un algoritmo polinómico. Y en ocasiones, hay un algoritmo polinómico para el de la palabra pero solo exponencial para el de la conjugación.

# Capítulo 2

# Grupos de trenzas

En este capítulo se define en primer lugar el grupo de trenzas $B_N$ mediante la presentación del grupo y su interpretación geométrica, de la cual deriva el nombre de *trenzas*. El diagrama obtenido geométricamente facilita identificar la estructura de grupo de $B_N$ así como su relación con el conjunto de permutaciones $S_N$ mediante un epimorfismo, lo cual permite caracterizar las trenzas.
Posteriormente, el capítulo se centra en las trenzas positivas, elementos del monoide $B_N^+$, donde las trenzas se escriben con letras de potencias positivas. El objetivo es demostrar que cada trenza de $B_N$ se puede expresar de manera única usando una forma canónica, conocida como la forma normal ponderada a la izquierda. Esta expresión única será muy relevante en los algoritmos criptográficos, ya que oculta parte de la palabra original de la trenza. Para ello, se estudiará primero en el caso de trenzas positivas, y a continuación, para todas las trenzas. Finalmente, se introduce una de las primera representaciones que se asoció al grupo de trenzas, la representación coloreada de Burau, que será de utilidad en el protocolo de Anshel-Anshel-Goldfeld (capítulo 3) y en la firma WalnutDSA (capítulo 4).

## 2.1. Definición de grupo de trenzas

El término de *grupo de trenzas* fue introducido por primera vez por el matemático Emil Artin en el año 1925 ([12]) aunque estos grupos ya habían sido considerados previamente (1891) por el matemático Hurwitz cuando estudió los grupos que en la terminología moderna se conocen como "grupos fundamentales de espacios de configuración de $n$ puntos en el plano complejo". Estos grupos juegan un papel importante en topología de baja dimensión, teoría combinatoria de grupos y en teoría de la representación.

En esta sección daremos una breve definición del grupo de trenzas $B_N$ que se formula en términos de la presentación del grupo a través de los generadores y sus relaciones, aunque la definición más gráfica e intuitiva es visualizar las trenzas como cuerdas que se entrelazan.

**Definición.** Sea $N$ un número natural. El *grupo de trenzas*, denotado $B_N$, es el grupo dado por la presentación finita $\langle X \mid R \rangle$ donde

$$\begin{aligned} X &= \{\sigma_1, \ldots, \sigma_{N-1}\} \\ R &= \Big\{ \sigma_i\sigma_j = \sigma_j\sigma_i \quad \text{para todo } i,j \in \{1,\ldots,N-1\} \text{ tal que } |i-j| \geq 2, \\ &\qquad \sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1} \quad \text{para } i \in \{1,\ldots,N-2\} \Big\} \end{aligned} \tag{2.1}$$

A los elementos $\sigma_i$ se les llama *generadores de Artin*.

Esta presentación fue obtenida por Artin en su artículo *Theory of Braids* en 1947 y será la que utilicemos en el presente trabajo. Sin embargo, no es la única, existen también otras presentaciones geométricas y topológicas [12].

Por definición, se deduce que:

- $B_1 = \{1\}$ es el grupo trivial.
- El grupo $B_2 = \langle \sigma_1 \rangle$ está generado por un único generador $\sigma_1$ y el conjunto vacío de relaciones, luego $B_2$ es el grupo cíclico infinito que es isomorfo al grupo libre $\mathbb{Z}$ mediante el isomorfismo $f : \mathbb{Z} \to G$ dado por $f(n) = a^n$.
- Los grupos $B_N$ con $N \geq 3$ se caracterizan por no ser grupos abelianos.

A los elementos de $B_N$ se les denominan *trenzas*, nombre que deriva de su interpretación geométrica.

**Interpretación geométrica**

Los elementos de $B_N$ se pueden ver como el movimiento de $N$ puntos de un plano superior (que en algunos artículos identifican con el plano $t = 0$) a un plano inferior paralelo a través de $N$ cuerdas o hebras (que son las $N-$trenzas). La cuerda $i-$ésima tiene un extremo en un punto $i$ del plano superior y el otro extremo en un punto $j$ del plano inferior, siendo $j = i$ o $j \neq i$, ya que puede o no cruzarse con otras cuerdas. Al igual que cada una de las cuerdas parte de un único punto, a cada posición $j = 1, \ldots, N$ del plano inferior llega solo una cuerda. A esta representación se le conoce como *diagrama de trenzas*.

En este diagrama, el generador de Artin $\sigma_i$ se representa como el cruce de la $i-$ésima cuerda con la $(i+1)-$ésima cuerda mientras que el resto de cuerdas quedan fijas ya que parten del punto $j \notin \{i, i+1\}$ y llegan al punto $j$ sin cruzar con ninguna otra cuerda. Para diferenciar la representación de $\sigma_i$ y la de su inverso, consideraremos que $\sigma_i$ cruza la cuerda $i$ por debajo de la cuerda $i+1$; en cambio en $\sigma_i^{-1}$ la cuerda $i$ cruza por encima de la cuerda $i+1$ [1].
En general, llamaremos *cruces positivos* cuando la cuerda que comienza en el punto $i$ cruza por debajo de la cuerda que comienza en el punto $j$ cuando $i < j$.


(a) Generador $\sigma_1$


(b) Trenza $AB \to A = \sigma_2$ y $B = \sigma_3$


(c) Trenza $BA \to A = \sigma_2$ y $B = \sigma_3$

El diagrama de Artin facilita la comprensión de la estructura de grupo de $B_N$, así como su relación con el grupo de permutaciones $S_N$. En primer lugar, veamos las propiedades del grupo:

[1] Algunos autores establecen el convenio al revés, de modo que $\sigma_i$ es la hebra $i$ cruzando por encima de la hebra $i+1$. Luego no hay un criterio definido, basta indicar previamente la orientación de la presentación del generador.

1. El producto de dos N-trenzas $\beta_1\beta_2$ se puede ver geométricamente de la siguiente forma: para representar el producto o composición de dos trenzas, se coloca primero la trenza $\beta_1$. Debajo, se hacen coincidir los puntos inferiores de $\beta_1$ con los puntos superiores de $\beta_2$, de manera que $\beta_1$ y $\beta_2$ quedan unidas entre sí (como en la imagen b) anterior). El resultado es la trenza $\beta_1\beta_2$.

   Por tanto, se cumple la propiedad **asociativa** $(\beta_1\beta_2)\beta_3 = \beta_1(\beta_2\beta_3)$. Sin embargo, la propiedad conmutativa no se cumple, ya que $\beta_1\beta_2 \neq \beta_2\beta_1$. Un ejemplo lo visualizamos en la imagen b) y c) donde efectivamente no se obtiene la misma trenza.

2. El elemento **neutro** existe, y se denomina N-trenza trivial $1_N$. Esta trenza deja fijas todas las cuerdas, es decir, la cuerda $i-$ésima parte del punto $i$ y llega al punto $i$ para todo $i = 1,\dots,N$, y no hay ningún cruce.

3. Toda N-trenza $\beta$ tiene un **inverso** $\beta^{-1}$, que también es una $N-$trenza tal que $\beta \circ \beta^{-1} = 1_N$. En la imagen c) anterior para la trenza $\beta = \sigma_3\sigma_2$ sería $\beta^{-1} = \sigma_2^{-1}\sigma_3^{-1}$.

Por tanto, $B_N$ es un grupo cuyos elementos son trenzas. Además, como $B_N$ es isomorfo al cociente de un grupo libre $F_N(X)$ con $X$ el conjunto de los generadores de Artin y de sus inversos (ver proposición 1.6), cada trenza $\beta \in B_N$ se puede escribir como una palabra de la forma

$$W_\beta = \sigma_{i_1}^{\varepsilon_1} \cdots \sigma_{i_n}^{\varepsilon_n} \quad \text{con} \ \ \varepsilon_i \in \{\ 1,-1\} \quad \text{para} \ \ 1 \leq i \leq N-1 \tag{2.2}$$

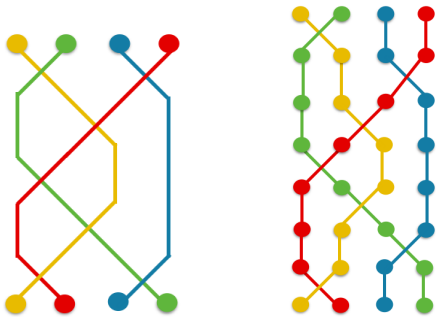
Dada esta expresión, se dirá que dos palabras $W_1, W_2$ son equivalentes si $W_1$ se puede transformar en $W_2$ (o viceversa) mediante las relaciones del conjunto $R$ de $B_N$. En caso de que sean dos palabras idénticamente iguales (iguales letra a letra) se denota $W_1 \equiv W_2$.

Por otro lado, este diagrama induce una permutación en el conjunto $\{1,\dots,N\}$. Así, se puede definir un homomorfismo de $B_N$ a $S_N$, donde $S_N$ es el grupo simétrico que contiene todas las permutaciones de $N$ elementos. En los generadores este homomorfismo viene dado por

$$\begin{aligned} \varphi : B_N &\to S_N \\ \sigma_i &\mapsto (i \ \ i+1) \end{aligned} \tag{2.3}$$

Luego, aplicando el homomorfismo $\varphi$ a la expresión (2.2) se obtiene una permutación en $S_N$ y se deduce que está bien definido ya que preserva las relaciones. Además, es fácil ver que $\varphi$ es suprayectivo y también que, si consideramos el conjunto de relaciones de la presentación de $B_N$ y se añade a dicho conjunto la relación $\sigma_i^2$ para $i = 1,\dots,N-1$ se obtiene una presentación del grupo $S_N$.

**Ejemplo 3.** Dada la representación de la trenza de la figura (a) que aparece a continuación, enumeramos de izquierda a derecha (la cuerda amarilla la 1 y la roja la 4) y visualizamos el diagrama. Se deduce facilmente que la permutación asociada $(2 \ \ 4)$.



(a) ⇒ (b)

Vamos a expresar la trenza de $(a)$ mediante los generadores. Para ello dividimos la trenza en varias partes, de modo que en cada división haya solo un cruce y le asignamos el generador correspondiente. Así obtenemos la representación (b) y escribimos, en orden, el generador que interviene en cada división (marcada por los puntos). Entonces:

una palabra que representa a la trenza es: $\sigma_1\sigma_3\sigma_2\sigma_1\sigma_2\sigma_3\sigma_1$

y la permutación inducida: $(1\ \ 2)(3\ \ 4)(2\ \ 3)(1\ \ 2)(2\ \ 3)(3\ \ 4)(1\ \ 2) = (1)(2\ \ 4)(3)$

Un razonamiento como el del ejemplo se puede aplicar para cualquier otro elemento de $B_N$. Es más, si por ejemplo, en el primer nivel se representa primero el cruce de la cuerda azul con la roja, tenemos una palabra diferente $\sigma_1\sigma_3\sigma_2\sigma_1\sigma_2\sigma_3\sigma_1$, pero la trenza subyacente es la misma y la permutación inducida también.

Recordar, además, que el kernel del epimorfismo $\varphi$ estará formado por todas las trenzas cuya permutación inducida es la identidad en $S_N$, lo cual motiva la siguiente definición:

**Definición.** Se dice *trenza pura* a la trenza cuya permutación inducida en el conjunto $\{1,\dots,N\}$ es la identidad. Es decir, que todas las hebras de la trenza comienzan y acaban en el mismo punto. Al grupo de trenzas puras se denota $P_N$ y es un subgrupo de $B_N$.

## 2.2. Trenzas positivas

Recordemos que un *monoide* $(A,\odot)$ es una estructura algebraica donde $A$ es un conjunto y $\odot : A\times A \to A$ una operación interna definida como $(a,b) \to c = a\odot b$ que es asociativa y tiene elemento neutro.

Las *trenzas o palabras positivas* se definen como trenzas que se escriben solo usando potencias positivas de los generadores de $\sigma_i$ para $i = 1,\dots,N-1$. En la presentación (2.1) de $B_N$, las relaciones se expresan en términos de los generadores con potencias positivas. Luego, podemos considerar en monoide $B_N^+$ dado por la misma presentación que $B_N$, cuyos elmentos son trenzas positivas para las cuales se puede establecer la siguiente relación de equivalencia si: dos trenzas son equivalentes si son idénticas o positivamente equivalentes, lo que significa que se puede transformar una en la otra aplicando reiteradamente las relaciones de $B_N$.

El monoide $B_N^+$ se denomina *monoide de las trenzas positivas*. De forma natural, se define el homomorfismo de monoides $B_N^+ \to B_N$, que Garside demostró que también es inyectivo [12].

Este resultado tiene una gran relevancia en el estudio de palabras equivalentes, de modo que dos trenzas positivas serán equivalentes en $B_N$, si lo son en $B_N^+$.

El monoide $B_N^+$ se caracteriza también por tener un orden parcial: Dadas dos palabras $W_1, W_2 \in B_N^+$, se dice que $W_1 \preceq W_2$, si existe alguna palabra $Y \in B_N^+$ tal que $W_1 Y = W_2$. En este caso decimos que $W_1$ es un *prefijo* de $W_2$. Además si $W_1 \preceq W_2$, entonces se cumple $XW_1 \preceq XW_2$, luego este orden parcial es invariante al multiplicar por la izquierda.
Análogamente, se puede definir el orden $\succeq$, donde diremos que $W_1$ es un *sufijo* de $W_2$ si.

En esta sección vamos a demostrar que las trenzas positivas admiten una factorización única que depende solo de términos positivos, conocida como la *factorización ponderada la izquierda*. Para entender esta descomposición, introduciremos previamente algunos conceptos.

**Definición.** Sea $P$ una palabra positiva, $P \in B_N^+$.
El *conjunto inicial* de $P$, denotado $S(P)$, es el conjunto de índices $i \subseteq \{\, 1, \ldots, N-1\}$ tal que

$$S(P) = \{\, i \mid P = \sigma_i P' \text{ para alguna trenza } P' \in B_N^+\}$$

El *conjunto final* de $P$, denotado $F(P)$, es el conjunto de índices $i \subseteq \{\, 1, \ldots, N-1\}$ tal que

$$F(P) = \{\, i \mid P = Q' \sigma_i \text{ para alguna trenza } Q' \in B_N^+\}$$

Se dice entonces que una factorización $P = AB$ con $A, B \in B_N^+$ es *ponderada a la izquierda* si se cumple que $S(B) \subset F(A)$ o *ponderada a la derecha* si $F(A) \subset S(B)$.

Como ya se ha estudiado anteriormente, existe una correspondencia entre las trenzas y las permutaciones que se puede utilizar para caracterizar los conjuntos inicial y final. Consideremos previamente la siguiente observación:

Suponer que tenemos una permutación tal que $\pi(i) = b_i$ para $i = 1, \ldots, N$ representando la $i$-ésima cuerda que comienza en el punto $i$ y acaba en el punto $b_i$. Vamos a ver que existe una trenza positiva cuya permutación asociada es $\pi$ y a expresarla en términos de los generadores de Artin (ver ejemplo 3). Se puede proceder de la siguiente forma:

1. Primero, unir cada uno de los puntos del plano superior con sus correspondientes imágenes por $\pi$ de forma que los cruces sean positivos. Después, dividir la figura en tantas partes como cruces haya, de modo que en cada una de las partes se produzca un cruce (entre dos hebras consecutivas $i$, $i+1$).

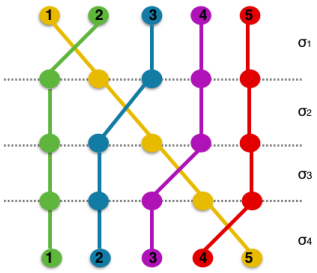
2. Asociar a dicho cruce el generador de Artin $\sigma_i$. Así, en cada subdivisión se aplica un generador cuya potencia es positiva (ya que hemos establecido que los cruces sean positivos).

3. Obtener la palabra que representa a $A_\pi$ concatenando los generadores (siguiendo el orden en el que están las subdivisiones).

Las trenzas $A_\pi$ obtenidas en este proceso se caracterizan por cumplir que cada par de cuerdas $(i, j)$ se cruza como máximo una vez, para $1 \leq i, j \leq N$. Estas trenzas $A_\pi$ se denominan *trenzas de permutación positiva* y al conjunto de todas ellas se denota $S_N^+$.

**Ejemplo 4.** Sea $\pi = (1\ 5\ 4\ 3\ 2)$ una permutación de $S_5$. Para llevar acabo la representación geométrica se consideran 5 hebras (diferenciadas por colores) de modo que $i \to b(i)$

$$1 \to 5 \quad 2 \to 1 \quad 3 \to 2 \quad 4 \to 3 \quad 5 \to 4$$




Se observan cuatro cruces y cada uno de ellos se asocia a un generador $\sigma_i$, que recordamos que permutaba las hebras $i$, $i+1$. De ello se deduce que la trenza correspondiente a $\pi$ es $A_\pi = \sigma_1\sigma_2\sigma_3\sigma_4$.

Estos diagramas son también de gran utilidad cuando se quiere obtener el conjunto inicial de una palabra $P$ como se enuncia en el siguiente lema.

**Lema 2.1.** *Sea $A_\pi \in S_N^+$ y $\pi \in S_N$ la permutación asociada. Las siguientes afirmaciones son equivalentes:*

1. $i \in S(A_\pi)$.
2. *La hebra $i$ cruza con la hebra $i+1$ en $A_\pi$.*
3. $\pi(i) > \pi(i+1)$, *es decir,* $b_i > b_{i+1}$.

*Demostración.* 1) $\Rightarrow$ 2): Si $i \in S(A_\pi)$ por definición existe $A' \in B_N^+$ tal que $A_\pi = \sigma_i A'$. Y como $\sigma_i$ es el generador que permuta la hebra $i$ con la $i+1$, eso implica que se cruzan, luego se cumple 2).
2) $\Rightarrow$ 3): Como $A_\pi \in S_N^+$ cada par de hebras solo se cruza una vez, luego si $i$ se cruza con $i+1$ lo que ocurre es que se aplica la trasposición $(i\ \ i+1)$ luego $\pi(i) > \pi(i+1)$.
3) $\Rightarrow$ 1): Si $\pi(i) > \pi(i+1)$ es porque la permutación $\pi$ afecta a esas dos hebras produciendo su cruce, ya que en caso contrario como $i+1 > i$, se mantendría $\pi(i+1) > \pi(i)$. Luego, debido a que solo se cruzan una vez, se puede dibujar el diagrama de manera que el primer cruce coincida exactamente con el cruce de $i$ con $i+1$. Luego, la palabra correspondiente a $A_\pi$ tiene como primer término a $\sigma_i$, es decir, $A_\pi = \sigma_i A'$, con $A' \in S_N^+$. □

Esta proposición caracteriza al conjunto $S(A_\pi)$, de manera que dicho conjunto se puede definir también como

$$S(A_\pi) = \{\, i \mid \pi(i) > \pi(i+1)\}.$$

Aplicando este lema a la trenza inversa se deduce la caracterización para el conjunto final

$$F(A_\pi) = \{\, i \mid \pi^{-1}(i) > \pi^{-1}(i+1)\}.$$

Sin necesidad de cálculo, solo observando el diagrama del ejemplo 4, se puede determinar el conjunto inicial y final, que son

$$S(A_\pi) = \{1\} \quad \& \quad F(A_\pi) = \{4\}.$$

**Proposición 2.2** (Ver [8], Lema 2.2, demostración p.9)**.** *Cada palabra positiva $P \in B_N^+$ tiene una única expresión ponderada a la izquierda de la forma $P = A_1P_1$ con $A_1 \in S_N^+$ tal que dada otra factorización $P = AB$ con $A \in S_N^+$ se cumple que $A_1 = AQ$ para alguna palabra $Q \in B_N^+$.*

**Corolario 2.3.** *Sea $P \in B_N^+$ una palabra positiva para la cual existe una expresión ponderada a la izquierda $P = A_1P_1$ tal que $A_1 \in S_N^+$ , $P_1 \in B_N$ y $S(P_1) \subseteq F(A_1)$. Entonces $S(A_1) = S(P)$.*

*Demostración.* Procedemos a demostrar la igualdad por doble contenido:

$S(A_1) \subset S(P)$: Suponer $i \in S(A_1)$, por la definición de conjunto inicial, $A_1 = \sigma_i Q_i$ donde $Q_i \in B_N^+$. Luego, $P = A_1P_1 = \sigma_i R_i$ donde $R_i = Q_iP_1 \in B_N^+$. Por tanto, $i \in S(P)$.

$S(P) \subset S(A_1)$: Suponer $i \in S(P)$, por definición, $P = \sigma_i B \equiv AB$ con $B \in B_N^+$. Por el lema 2.2, la igualdad anterior se cumple solo si $A_1 = \sigma_i Q \equiv AQ$ con $Q \in B_N^+$, es decir, si $i \in S(A_1)$. □

**Teorema 2.4.** *Sea $P \in B_N^+$ una palabra positiva. Existe una única P ponderada a la izquierda, $P = A_1A_2 \cdots A_k$, con $A_i \in S_N^+$, donde $A_k$ es distinta de la palabra identidad, y $S(A_{i+1}) \subset F(A_i)$ para todo $1 \leq i \leq k$.*

*Demostración.* Denotar $P = P_0$. Por el lema 2.2, sabemos que existe una única factorización $P_0 = A_1P_1$ donde $A_1 \in S_N^+, P_1 \in B_N^+$. Aplicando de nuevo 2.2, se puede escribir $P_1 = A_2P_2$ con $A_2 \in S_N^+, P_2 \in B_N^+$ de manera única. Luego, sustituyendo $P_1$ se obtiene $P_0 = A_1A_2P_2$ donde $A_1, A_2 \in S_N^+$ y $P_2 \in B_N^+$.
Notar, además, que al factorizar una palabra $P_{i-1} = A_iP_i$ se cumple que $A_i$ es al menos una palabra de longitud uno, por lo que la longitud de $P_i$ es estrictamente menor que la de $P_{i-1}$. Luego el proceso de aplicar reiteradamente el lema 2.2 es finito y acaba cuando $P_{k-1} = A_kP_k$ donde $P_k$ la palabra vacía.

Así, tras factorizar cada una de las palabras $P_i$ con $i = 1, \ldots, k-1$ y sustituirlas en la expresión $P_0 = A_1P_1$ se obtiene

$$P = A_1A_2 \cdots A_k \quad \text{con } A_i \in S_N^+ \text{ para } i = 1, 2, \ldots, k.$$

Además, como las expresiones $P_{i-1} = A_iP_i$ son ponderadas a la izquierda se cumple, por definición, que $S(P_i) \subset F(A_i)$. Por el corolario 2.3 se sabe que, $S(A_{i+1}) = S(P_i)$. Por tanto,

$$S(A_{i+1}) = S(P_i) \subset F(A_i) \text{ para todo } 1 \leq i \leq k.$$

□

Este teorema demuestra la existencia y unicidad de una expresión ponderada a la izquierda, pero no determina un algoritmo concreto sobre cómo se puede obtener dicha factorización. Dada una palabra $W$ positiva, algunos autores ([12], [8]) proponen

1. Dividir la palabra $W$ en bloques de subpalabras (se permiten bloques de longitud uno) que representen trenzas de permutación positivas, evitando así que en un bloque haya una potencia de $\sigma_i$, $i = 1, \ldots, N-1$.

$$W = (w_1)(w_2) \cdots (w_k)$$

2. Observar dos bloques consecutivos y comprobar si algún término de $w_{i-1}$ se puede pasar al bloque $w_i$ de modo que $w_{i-1}$ siga cumpliendo que cada par de hebras cruza como máximo una vez. En caso afirmativo, pasar ese término al bloque $w_i$ teniendo en cuenta las relaciones de $B_N$.

3. Repetir el paso anterior hasta que no se puedan mover más términos de un bloque a otro.

## 2.3. Forma normal ponderada a la izquierda

La forma normal es una forma canónica de representar cada uno de los elementos de un grupo, de manera que cada elemento solo tiene una forma normal y en caso de que existan dos elementos con la misma forma normal, han de ser iguales. Existen varias formas normales para los grupos de trenzas, en esta sección introduciremos la forma normal ponderada a la izquierda, teniendo en cuenta los resultados presentados sobre trenzas positivas. Esta forma canónica es una versión mejorada de la forma normal que descubrió Garside, la cual introducimos previamente junto con el nuevo concepto de trenza fundamental, así como el mínimo común múltiplo en $B_N^+$ de dos generadores $\sigma_i$, $\sigma_j$ definido como

$$\sigma_i \vee \sigma_j = \begin{cases} \sigma_i\sigma_j & \text{si } |i-j|>1 \\ \sigma_i\sigma_j\sigma_i & \text{si } |i-j|=1 \end{cases}$$

Es fácil demostrar que esta operación es asociativa. Basta considerar todos los casos posibles y ver que efectivamente coincide $(\sigma \vee \sigma_j) \vee \sigma_k = \sigma_i \vee (\sigma_j \vee \sigma_k)$.

**Definición.** Una trenza de $B_N$ se dice *trenza fundamental* y se denota $\Delta_N$ si

$$\Delta_N = \Delta_{N-1}\sigma_{N-1}\sigma_{N-2}\cdots\sigma_1$$

donde $\Delta_1 = \sigma_1$. De esta fórmula recursiva se deduce la siguiente expresión

$$\Delta_N = \sigma_1(\sigma_2\sigma_1)\cdots(\sigma_{N-1}\cdots\sigma_1) \tag{2.4}$$

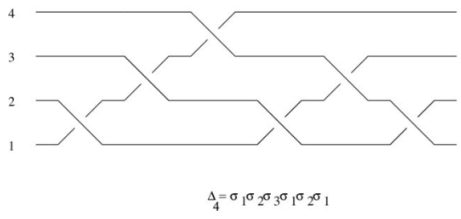



Figura 2.1: Trenza fundamental de 4 hebras, $\Delta_4$.

En la figura 2.1 se visualizan fácilmente las cuatro hebras y los cruces positivos, de manera que cada par de hebras se cruza una vez. Esto se cumple para todo $N$, ya que las trenzas fundamentales son trenzas de permutación positiva cuya permutación $\pi$ tal que $\pi(i) = b_i$ es exactamente

$$\varphi(\Delta_N) = (1\ \ N)(2\ \ N-1)\cdots([N/2]\ \ [N/2]+1). \tag{2.5}$$

siendo $b_i$ la posición final a la que llega la hebra $i$. Esta afirmación se puede demostrar procediendo por inducción aplicando el homomorfismo definido en (2.3). En caso de que $N$ sea

impar, la última permutación será simplemente $(\lceil N/2 \rceil)$. Esta observación permite identificar rápidamente el conjunto inicial y final (ver p.16) de la palabra positiva $W_{\Delta_N}$ que representa a $\Delta_N$.
En lo que sigue se presentan todos los resultados para el caso genérico de $B_N$, denotaremos $\Delta = \Delta_N$ sobreentendiendo el valor de $N$ en el subíndice.

**Proposición 2.5.** *Sea $\Delta$ la trenza fundamental de N cuerdas. Entonces, $S(\Delta) = F(\Delta) = \{1,\dots,N-1\}$.*

*Demostración.* Por el lema 2.1, $i \in S(W_\Delta)$ si $b_i = \pi(i) > \pi(i+1) = b_{i+1}$. En (2.5) se observa que $\pi(i) = (N+1) - i$, luego $\pi(i) > \pi(j)$ para todo $i < j$ tal que $1 \leq i, j \leq N$. Por tanto,

$$S(\Delta) = \{1,\dots,N-1\}.$$

Análogamente, como $\pi^{-1}(i) = (N+1) - i$, se cumple $\pi^{-1}(i) > \pi^{-1}(j)$ para $i < j$. Por tanto,

$$F(\Delta_N) = \{1,\dots,N-1\}.$$

□

**Proposición 2.6.** *Sea $A \in S_N^+$ tal que $S(A) = \{1,\dots,N-1\}$. Entonces $A = \Delta$.*

*Demostración.* Sea $\pi$ la permutación asociada a $A \in S_N^+$. Dado el conjunto $S(A)$ y aplicando la definición, se tiene que $\pi(i) > \pi(i+1)$ para cada $i$. Es más, $\pi(i) > \pi(j)$ para todo $i < j$, lo cual significa que cada par de hebras $(i, j)$ se cruza al menos una vez en $A$. Por hipótesis, $A \in S_N^+$, es decir, $A \in B_N^+$ y cada par de hebras cruza máximo una vez. Luego $A = \Delta$. □

A lo largo del trabajo, usaremos la expresión (2.4) de $\Delta$. Sin embargo, esta no es única y en muchos artículos se puede encontrar escrita como una palabra equivalente

$$\Delta_N = (\sigma_1 \cdots \sigma_{N-1})(\sigma_1 \cdots \sigma_{N-2}) \cdots (\sigma_1).$$

Precisamente esta flexibilidad a la hora de expresarse implica la siguiente proposición:

**Proposición 2.7.** *La trenza fundamental verifica las siguientes propiedades:*

1. $\Delta = \sigma_1 \vee \sigma_2 \vee \cdots \vee \sigma_{N-1}$.
2. $\sigma_i \Delta = \Delta \sigma_{N-i}$ *para* $1 \leq i \leq N-1$. (Ver [10], Lema 1)
3. $\Delta^2$ *conmuta con todo elemento de* $B_N^+$, *es decir,* $\Delta^2$ *genera el centro de* $B_N^+$. (Ver [10], Teorema 7)
4. *Existen* $X_i, Y_i \in B_N^+$ *tales que* $\sigma_i X_i = \Delta = Y_i \sigma_i$ *para* $1 \leq i \leq N-1$, *es decir,* $\sigma_1, \dots, \sigma_{N-1}$ *son sufijos y prefijos de* $\Delta$. (Ver [10], Lema 4)

**Proposición 2.8.** *Para toda trenza $\beta \in B_N^+$ se tiene $1_N \preceq \beta \preceq \Delta^r$ para $r \in Z$.*

*Demostración.* En primer lugar, por la definición de trenza fundamental y recordando como se define el orden parcial, se cumple:

$$1 \preceq \sigma_i \preceq \Delta \quad i = 1,\dots,N-1.$$

donde $1_N$ es la trenza identidad. Además, como $\beta \in B_N^+$ sabemos que no existen potencias negativas, por lo cual, cada uno de los generadores será prefijo de $\Delta$ y tendrá al elemento $1_N$ como prefijo. Entonces, como toda trenza de $B_N^+$ se puede expresar como una palabra cuyas letras son potencias de $\sigma_1, \dots, \sigma_{N-1}$, entonces

$$1_N \preceq \beta \preceq \Delta^r \quad r \in \mathbb{Z}.$$

□

**Observación.** En el caso de las trenzas de permutación positivas $A_\pi$, se cumple también que

$$1_N \preceq A_\pi \preceq \Delta^r \quad r \in \mathbb{Z}.$$

Como los cruces en $A_\pi$ son positivos, se deduce que $r$ es no negativa. Además, cada par de hebras se cruza como máximo una vez, lo que implica que y $r = 1$.

Teniendo en cuenta esta proposición, si consideramos una trenza escrita como una palabra en $\sigma_i, \ldots, \sigma_{N-1}$ y sus inversos, usando la propiedad 5, cada letra $\sigma_i^{-1}$ se puede reemplazar por $\Delta^{-1}X_i$. Además, por 2. se cumple $\sigma_i\Delta^{-1} = \Delta^{-1}\sigma_{N-i}$, luego cada subpalabra de la forma $\sigma_i\Delta^{-1}$ la podemos sustituir por su expresión equivalente, desplazando el término $\Delta^{-1}$ un puesto hacia la izquierda. Luego, cada trenza puede ser escrita como $\Delta^p A$ donde $p \in \mathbb{Z}$ y $A \in B_N^+$.

Observar además, que si $\Delta \preceq A$ entonces se podría reducir la longitud de $A$, ya que bastaría sustituir $\Delta^p$ por $\Delta^{p+1}$ y $A$ por $\Delta^{-1}A$. Luego este proceso solo se puede hacer un número finito de veces. Por tanto, $\Delta \not\preceq A$. Tiene sentido entonces la siguiente definición:

**Definición.** Sea $W \in B_N$ una palabra. La *forma normal de Garside* se define como la factorización $W = \Delta^p A$ donde $p \in \mathbb{Z}$ y $A \in B_N^+$ con $\Delta \not\preceq A$.

En el teorema 2.4 se ha demostrado que para cada palabra de $B_N^+$ existe una factorización única. Luego se puede concretar un poco más la definición anterior con el siguiente teorema:

**Teorema 2.9** (Forma normal ponderada a la izquierda)**.** *Sea una trenza $\beta \in B_N$ y una palabra $W_\beta$ que representa a $\beta$. Entonces, $W_\beta$ se expresa de manera única como*

$$W_\beta = \Delta^p A_1 A_2 \cdots A_k$$

*donde $p \in \mathbb{Z}$ y las trenzas de permutación $1 \prec A_i \prec \Delta$ verifican que $A_iA_{i+1}$ es una subpalabra ponderada a la izquierda, para $1 \leq i \leq k$.*

*Demostración.* Suponer que $W_\beta$ es una palabra positiva. Por el teorema 2.4 existe una única factorización ponderada a la izquierda:

$$W_\beta = A_1 \cdot A_2 \cdots A_k$$

donde $A_i \in S_N^+$ y $S(A_{i+1}) \subset F(A_i)$. Entonces si:

- Si una potencia de $\Delta$ es prefijo de $W_\beta$, ha de ser una potencia positiva, luego $\Delta$ es prefijo. Esto implica que $S(W_\beta) = \{1, \ldots, N-1\}$. Usando el corolario 2.3 se deduce que $S(A_1) = \{1, \ldots, N-1\}$ y por la proposición 2.6, $A_1 = \Delta$.
  Ahora, si $\Delta^2$ fuera prefijo de $W_\beta$ repitiendo el mismo argumento de arriba se llegaría a $A_2 = \Delta$. Y esto es una contradicción, ya que $S(A_2) \subset F(A_1)$.
- Si una potencia de $\Delta$ no es prefijo de $W_\beta$, entonces se verifica la tesis para $p = 0$.

Suponer ahora que $W_\beta \in B_N$. Bastará demostrar que $W_\beta$ se puede escribir en la forma normal de Garside, es decir, $W_\beta = \Delta^p A$ con $A \in B_N^+$, puesto que si existe $A \in B_N^+$, entonces por 2.4 existe una factorización única.

Escribimos escribir $W_\beta$ como una palabra de la forma

$$W_\beta = W_1 \cdot x_1^{-1} \cdot W_2 \cdot x_2^{-1} \cdots W_s \cdot x_s^{-1} W_{s+1}$$

donde $W_j$ es una palabra positiva de longitud $\geq 0$ y $x_j$ representa a un generador $\sigma_i$ con $1 \leq j \leq s$, $1 \leq i \leq N-1$.
Por la proposición 2.7, se cumple que para cada uno de los generadores existe una palabra positiva $X_i$ tal que $\Delta = \sigma_i X_i$ , luego $(\sigma_i)^{-1} = X_i \Delta^{-1}$, luego como cada $x_j$ es un generador,

$$W = W_1(X_1)\Delta^{-1}W_2(X_2)\Delta^{-1}\cdots W_s(X_s)\Delta^{-1}W_{s+1}.$$

Finalmente, por las propiedades 2.7, se pueden "mover" los factores $\Delta^{-1}$ a la izquierda para conseguir la expresión $W = \Delta^{-t}A$. Aplicando el teorema 2.4 a la palabra $A \in B_N^+$, se tiene la única factorización ponderada a la izquierda $A = A_1 \cdots A_k$ con $m \in \{0,1\}$.
Razonando como en (1), si $\Delta$ no es prefijo de $A$ entonces

$$W = \Delta^{-t}A_1 \cdot A_2 \cdots A_k = \Delta^p A_1 \cdots A_k.$$

Se cumple la tesis para $p = -t \in \mathbb{Z}$. Y si $\Delta$ es prefijo de $A$ entonces

$$W = \Delta^{-t+1}A_2 \cdots A_k = \Delta^p A_2 \cdots A_k.$$

Se cumple la tesis para $p = -t+1 \in \mathbb{Z}$.

Por último, para demostrar la unicidad, se consideran dos expresiones en la forma normal de Garside $\Delta^p A = W = \Delta^m B$, donde $A, B \in B_N^+$ no tienen a $\Delta$ como prefijo (ya que sino, por lo razonado anteriormente se reagrupo en la parte izquierda de la palabra).
Suponer primero el caso $p < m$, entonces $A = \Delta^{m-p}B$. Pero por lo que acabamos de razonar, esto no es cierto. Y para el caso $m < p$ ocurre lo mismo.
Luego $p = m$, luego $A = B$ en $B_N^+$, luego son positivamente equivalentes y por la proposición 2.4, tienen la misma factorización ponderada a la izquierda. Queda así demostrada la unicidad. □

**Nota.** A menudo, para una palabra $W = \Delta^p A_1 \cdots A_k$ en su forma normal se usa la notación $(p, \pi_1, \ldots, \pi_k)$ donde $p$ es el exponente de la trenza fundamental y $\pi_i$ es la permutación inducida por la palabra positiva $A_i$ para $i = 1, \ldots, k$.

En los dos ejemplos siguientes se calcula de la forma normal ponderada a la izquierda siguiendo los pasos de la demostración 2.3 y la propuesta comentada tras el teorema 2.4. En el primero de ellos, se muestra también el procedimiento a seguir simplemente observando el diagrama de Artin.

**Ejemplo 5.** Considerar el grupo $B_4$. Sea $\beta = \sigma_3\sigma_1\sigma_2^{-1}\sigma_3\sigma_2\sigma_3 \in B_4$.

1. Sustituir el factor $\sigma_2^{-1}$: Sea $\Delta_4 = (\sigma_1)(\sigma_2\sigma_1)(\sigma_3\sigma_2\sigma_1)$. Usando las relaciones del monoide $B_N^+$,

$$\begin{aligned}\sigma_1(\sigma_3\sigma_2\sigma_1) &= \sigma_1(\sigma_3\sigma_2)(\sigma_2\sigma_1) = (\sigma_3\sigma_2)(\sigma_1\sigma_2\sigma_1) \\ &= (\sigma_3\sigma_2)(\sigma_2\sigma_1\sigma_2) = (\sigma_3\sigma_2)(\sigma_2\sigma_1)(\sigma_2) = (\sigma_3\sigma_2\sigma_1)\sigma_2.\end{aligned}$$

   Luego,

$$\Delta_4 = (\sigma_1)(\sigma_2)(\sigma_3\sigma_2\sigma_1)(\sigma_2) \longrightarrow \sigma_2^{-1} = \Delta_4^{-1}(\sigma_1)(\sigma_2)(\sigma_3\sigma_2\sigma_1).$$

   Reescribimos,

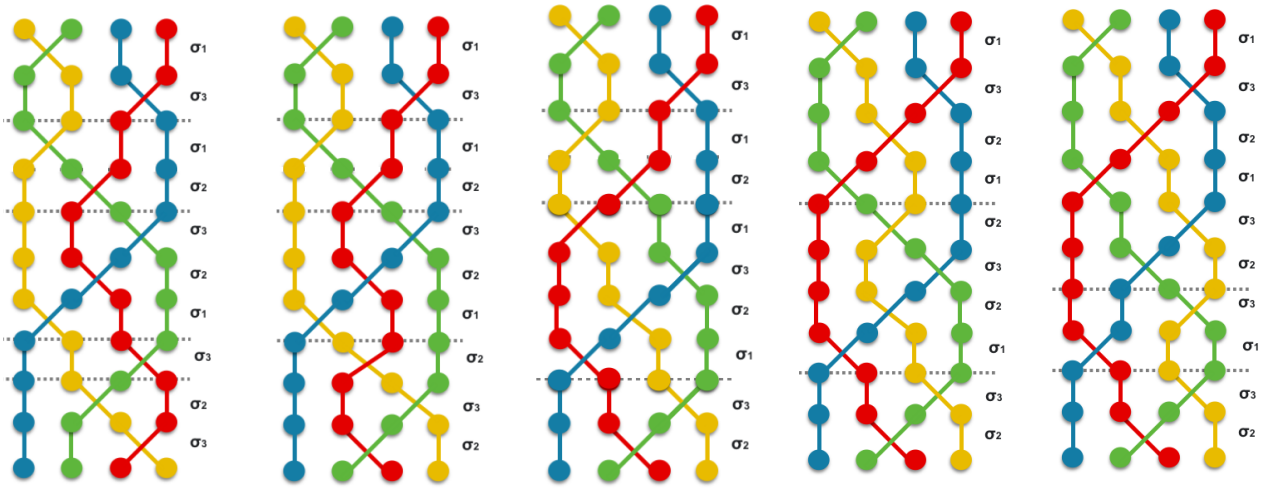
$$\beta = \sigma_3\sigma_1[\Delta_4^{-1}(\sigma_1)(\sigma_2)(\sigma_3\sigma_2\sigma_1)]\sigma_3\sigma_2\sigma_3$$

2. Agrupar las potencias de $\Delta_4$ a la izquierda usando: $\sigma_1\Delta_4^{-1} = \Delta_4^{-1}\sigma_3$, $\sigma_3\Delta_4^{-1} = \Delta_4^{-1}\sigma_1$.

$$\beta = \Delta_4^{-1}\sigma_1\sigma_3(\sigma_1)(\sigma_2)(\sigma_3\sigma_2\sigma_1)\sigma_3\sigma_2\sigma_3 \equiv \Delta_4^{-1}A$$

3. Calcular la forma normal ponderada a la izquierda para la palabra positiva $A \in B_4^+$:

$$\begin{aligned}
A &= (\sigma_1\sigma_3)(\sigma_1\sigma_2)(\sigma_3\sigma_2\sigma_1)(\sigma_3)(\sigma_2\sigma_3) && \leftarrow \text{El bloque 5 puede pasar al 4}\\
&= (\sigma_1\sigma_3)(\sigma_1\sigma_2)(\sigma_3\sigma_2\sigma_1)(\sigma_3\sigma_2\sigma_3) && \leftarrow \text{En el bloque 4 usar } \sigma_2\sigma_3\sigma_2 = \sigma_3\sigma_2\sigma_3\\
&= (\sigma_1\sigma_3)(\sigma_1\sigma_2)(\sigma_3\sigma_2\sigma_1)(\sigma_2\sigma_3\sigma_2) && \leftarrow \text{Se puede pasar } \sigma_2 \text{ al bloque 3}\\
&= (\sigma_1\sigma_3)(\sigma_1\sigma_2)(\sigma_3\sigma_2\sigma_1\sigma_2)(\sigma_3\sigma_2) && \leftarrow \text{Por la propiedad 3. de 2.7, } (\sigma_3\sigma_2\sigma_1)\sigma_2 = \sigma_1(\sigma_3\sigma_2\sigma_1)\\
& && \qquad \text{y pasar el bloque 3 al 2}\\
&= (\sigma_1\sigma_3)(\sigma_1\sigma_2\sigma_1\sigma_3\sigma_2\sigma_1)(\sigma_3\sigma_2) && \leftarrow \text{Usar } \sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2\\
&= (\sigma_1\sigma_3)(\sigma_2\sigma_1\sigma_2\sigma_3\sigma_2\sigma_1)(\sigma_3\sigma_2) && \leftarrow \text{Pasar } \sigma_2\sigma_1 \text{ al bloque 1 y usar } \sigma_3\sigma_2\sigma_3 = \sigma_2\sigma_3\sigma_2\\
&= (\sigma_1\sigma_3\sigma_2\sigma_1)(\sigma_3\sigma_2\sigma_3\sigma_1)(\sigma_3\sigma_2) && \leftarrow \text{Pasar } \sigma_3\sigma_2 \text{ al bloque 1}\\
&= (\sigma_1\sigma_3\sigma_2\sigma_1\sigma_3\sigma_2)(\sigma_3\sigma_1)(\sigma_3\sigma_2) && \leftarrow \text{Fin}
\end{aligned}$$



1º 2º 3º 4º 5º

Se puede comprobar que efectivamente:

$$\{1,3\} = S(A_2) \subset F(A_1) = \{1,2,3,4\} \quad \&\{3\} = S(A_3) \subset F(A_2) = \{1,3\}.$$

La forma normal ponderada a la izquierda es:

$$\beta = \Delta_4^{-1}A_1A_2A_3$$

con $A_1 = \sigma_1\sigma_3\sigma_2\sigma_1\sigma_3\sigma_2$, $A_2 = \sigma_3\sigma_1$ y $A_3 = \sigma_3\sigma_2$.

Este ejemplo representa la forma práctica en la que llevar acabo el algoritmo de encontrar la forma normal ponderada a la izquierda. Sin embargo, este método fue descrito explícitamente por El-rifai y Morton de la siguiente forma: Se define $d = a \wedge b$ define el máximo común divisor entre $a$ y $b$ tal que, teniendo el cuenta el orden parcial definido para $B_N^+$ se cumple $d \preceq a$, $d \preceq b$, $d' \preceq d$ para todo prefijo $d'$ común de $a$ y $b$. Entonces,

1. Sea $\beta \in B_N$ con su forma normal de Garside $\Delta^p A$.
2. Definir $A_1 = A \wedge \Delta$.
3. Computar $A_i = (A_{i-1}^{-1} \cdots A_1^{-1} A) \wedge \Delta, \ \ \forall\, i > 1$.
4. La forma normal ponderada a la izquierda es $\Delta^p A_1 \cdots A_k$.

## 2.4. Representación coloreada de Burau

Werner Burau fue el primero que estudió una representación del grupo de trenzas, a la que posteriormente se le dio el nombre con el que se conoce actualmente, *representación de Burau* ([3],[15] [2]). Esta representación será de interés tanto en el protocolo de Anshel-Anshel-Goldfeld (sección 3.3.2) como en la firma digital Walnut DSA (capítulo 4).

Sea $\{t_1, \ldots, t_N\}$ una lista de variables y $\mathbb{Z}[t_i^{\pm 1}]$ el anillo de polinomios de Laurent $f(t_i)$ en las variables $t_1, \ldots, t_N$, es decir, para $i = 1, \ldots, N$:

$$f(t_i) = a_k t_i^k + a_{k+1} t_i^{k+1} + \cdots + a_m t_i^m \ \text{ donde } \ a_j \in \mathbb{Z} \ \text{ para } \ j = k, \ldots, m \ \ (\text{con } k \leq m \in \mathbb{Z}).$$

Por otro lado, sea $GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}])$ el grupo de matrices $N-1 \times N-1$ invertibles en $\mathbb{Z}[t_i^{\pm 1}]$, es decir, las matrices cuyo determinante es $\varepsilon \cdot t_i^m$ para algún entero $m$ y $0 \neq \varepsilon \in \mathbb{Z}$.

**Definición.** Una *matriz coloreada de Burau*, denotada CB, es una matriz de $GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}])$ que se define para cada uno de los generadores de Artin de la siguiente manera [3]:

Para $i = 1$ :

$$CB(\sigma_1) = \left(\begin{array}{cc|c} -t_1 & 1 & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & I_{N-3} \end{array}\right) \qquad CB(\sigma_1^{-1}) = \left(\begin{array}{cc|c} -\frac{1}{t_2} & \frac{1}{t_2} & 0 \\ 0 & 1 & 0 \\ \hline 0 & 0 & I_{N-3} \end{array}\right)$$

Para $i = 2, \ldots, N-1$ :

$$CB(\sigma_i) = \left(\begin{array}{c|ccc|c} I_{i-2} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & t_i & -t_i & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & I_{N-i-2} \end{array}\right) \qquad CB(\sigma_i^{-1}) = \left(\begin{array}{c|ccc|c} I_{i-2} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & -\frac{1}{t_{i+1}} & \frac{1}{t_{i+1}} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & I_{N-i-2} \end{array}\right)$$

donde las variables aparecen en la fila $i$ para $2 \leq i \leq N-1$, y en el caso de $i = 1$, las variables aparecen en la primera fila y se omite una de las copias de la variable $t_1$.

Por otro lado, recordar el epimorfismo $\varphi : B_N \to S_N$ descrito en (2.3) de manera que a cada generador de Artin le podemos asociar una permutación $(i \ \ i+1)$ a la que denotaremos a partir de ahora $\alpha_i$. En el caso del inverso de un generador, la permutación es la misma, ya que es una trasposición. Luego, a cada generador de Artin se puede asociar una tupla $(CB(\sigma_i), \alpha_i)$

denominada *par coloreado de Burau*.

Dados dos pares coloreados de Burau, lo que nos interesa es poder definir una multiplicación entre ellos. Observar que el grupo de permutaciones $S_N$ actúa sobre $\mathbb{Z}[t_1^{\pm 1},\dots,t_N^{\pm 1}]$, de modo que si tenemos un polinomio en las variables $f(t_1,\dots,t_N)$, una permutación $\alpha \in S_N$ puede actuar (por la izquierda) permutando el índice de las variables. Esta acción se denota $f \mapsto {}^{\alpha} f$ tal que

$$ {}^{\alpha} f(t_1,t_2,\dots,t_N) = f(t_{\alpha(1)},\dots,t_{\alpha(N)}).$$

Esta acción se puede extender a $GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}])$ aplicándola a cada una de las entradas de la matriz. Dada una matriz $M = (f_{ij})$ cuyas entradas $f_{ij}$ son polinomios de Laurent, la acción se denota $M \mapsto {}^{\alpha} M$ y se define como

$$ {}^{\alpha} M = M({}^{\alpha} f_{ij}).$$

Teniendo en cuenta esta observación ya podemos definir formalmente el grupo coloreado de Burau con su operación interna:

**Definición.** El *grupo coloreado de Bureau*, denotado $CB_N$, es el producto semidirecto de $S_N$ y $GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}])$

$$ GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}]) \rtimes S_N = \{(M,\alpha) \mid M \in GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}]), \alpha \in S_N\}$$

donde $GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}])$ es el subgrupo normal en el que actúa $S_N$. El producto semidirecto es un grupo con la operación de la multiplicación (denotada $\cdot$) de dos pares coloreados de Burau que definimos a continuación:

$$ (M_1,\pi_1)\cdot(M_2,\pi_2) := (M_1\ {}^{\pi_1}M_2,\ \pi_1\pi_2).$$

En particular, el producto de los pares coloreados de Burau asociados a dos generadores de Artin $\sigma_i^{\pm 1}\cdot\sigma_j^{\pm 1}$ es [2]

$$ (CB(\sigma_i^{\pm 1}),\ \alpha_i)\cdot(CB(\sigma_j^{\pm 1}),\ \alpha_j) = (CB(\sigma_i^{\pm 1})\cdot{}^{\alpha_i} CB(\sigma_j^{\pm 1}),\ \ \alpha_i\cdot\alpha_j)$$

Así, hemos asociado un elemento del grupo coloreado de Burau a cada uno de los generadores de Artin. No obstante, recordar que $\beta \in B_N$ se puede expresar como $\beta = \sigma_{i_1}^{\varepsilon_1}\sigma_{i_2}^{\varepsilon_2}\cdots\sigma_{i_k}^{\varepsilon_k}$ con $\varepsilon_j \in \{\pm 1\}$. Luego podemos extender esta definición para toda trenza de $B_N$, como sigue:

El par coloreado de Burau para la trenza $\beta$ se define como

$$ (CB(\beta),\alpha_\beta) = \big(CB(\sigma_{i_1}^{\varepsilon_1})\cdot{}^{\alpha_{i_1}} CB(\sigma_{i_2}^{\varepsilon_2})\cdot{}^{\alpha_{i_1}\alpha_{i_2}} CB(\sigma_{i_3}^{\varepsilon_3})\ \cdots\ {}^{\alpha_{i_1}\alpha_{i_2}\cdots\alpha_{i_{k-1}}} CB(\sigma_{i_k}^{\varepsilon_k}),\ \ \sigma_{i_1}\sigma_{i_2}\cdots\sigma_{i_k}\big). \tag{2.6}$$

Quedan así definidos todos los conceptos que se necesitan para formalizar la siguiente definición:

**Definición.** La *representación coloreada de Burau*, denotada $\prod_{CB}$ se define como la aplicación

$$\begin{aligned} \prod\nolimits_{CB} : B_N &\longrightarrow CB_N \\ \beta &\longmapsto (CB(\beta),\alpha_\beta). \end{aligned}$$

[2] En algunos artículos, se denota el producto al revés y en consecuencia el producto se define como $(\pi_1,M_1)\cdot(\pi_2,M_2) = (\pi_1\pi_2,{}^{\pi_2^{-1}}(M_1)M_2)$. El motivo de invertir la permutación $\pi_2$ es para que la operación sea asociativa.

Además, se puede comprobar que $\prod_{CB}$ verifica las relaciones de $B_N$, luego

**Proposición 2.10.** *La representación $\prod_{CB} : B_N \to CB_N$ es un homomorfismo de grupos.*

Notar que por definición $CB_N$ es un grupo con la multiplicación entre dos pares coloreados de Burau. Es fácil ver que:

- $(I_{N-1}, I_{S_N})$ es la identidad.
- $(M,\alpha)^{-1} = ({}^{\alpha}M^{-1}, \alpha^{-1})$ es el elemento inverso de un elemento $(M,\alpha) \in CB_N$.

Posteriormente se puede probar que $\prod_{CB}$ verifica las relaciones de $B_N$. Para ello, se usará la definición de par coloreado de Burau en (2.6), y teniendo en cuenta que $\sigma_i\sigma_j = \sigma_j\sigma_i$ para $|i-j| \geq 2$ y que $\sigma_i\sigma_{i+1}\sigma_i = \sigma_{i+1}\sigma_i\sigma_{i+1}$ para $i = 1,\ldots,N-1$, se obtiene el resultado.

De esta proposición se deduce que la representación coloreada de Burau es una representación del grupo de trenzas $B_N$. Esta representación será un elemento importante en la descripción del protocolo de Anshel-Anshel-Goldfeld definido en el siguiente capítulo y en la firma digital WanutDSA donde se usa para generar la clave pública del protocolo.

# Capítulo 3

# Problemas algorítmicos en grupos de trenzas

La motivación del trabajo se presenta en este capítulo. En él, se describen algunos de los problemas de teoría combinatoria y computacional de grupos que han surgido a raíz de la criptografía. La criptografía de clave pública nace de la mano de Diffie y Hellman ([23], [11]) que sugirieron poder transmitir información confidencial usando solo canales públicos, es decir, sin necesidad de acordar canales secretos. Su esquema se plantea usando como grupo plataforma los grupos cíclicos infinitos. En las secciones 3.2 y 3.3 se describirán el problema de la palabra y el problema de la conjugación junto a dos sistemas criptográficos basados en ellos. En esta última sección, se enuncian también otros problemas de la teoría combinatoria de grupos como son: el problema del isomorfismo, el problema de descomposición o el problema de la pertenencia.

Cada uno de estos problemas se puede considerar como un problema de los siguientes tipos:

- Problema de decisión: Sea $\mathscr{P}$ una propiedad y **E** un elemento. Decidir si el elemento **E** verifica la propiedad $\mathscr{P}$ [1].
- Problema de búsqueda: Sea $\mathscr{P}$ una propiedad para la cual se sabe que existen elementos que la verifican. Encontrar al menos un elemento **E** que cumple $\mathscr{P}$.

## 3.1. Protocolo de Diffie-Hellman

El protocolo de intercambio de claves de Diffie-Hellman fue desarrollado en 1976 y fue el primer método que estableció un intercambio de claves a través de un canal de comunicación público. El esquema que propusieron permitía que dos usuarios pudiesen compartir información confidencial usando únicamente canales públicos, es decir, sin necesidad de acordar claves o intercambiar información secreta a priori [23].

En 2002, Hellman sugirió que el protocolo se llamase *intercambio de claves de Diffie-Hellman-Merkle* en honor a Ralph Merkle por su colaboración.

[1] En teoría computacional, se entiende por problema de decisión a un algoritmo que dado un alfabeto $X$ (por ejemplo, $X$ pueden ser los elementos del grupo en el que se aplica el problema), devuelve una respuesta para cada uno de los $x \in X$ de si o no, en función de si el elemento verifica o no la propiedad.

Presentamos en primer lugar el problema en el que se basa dicho protocolo.

*Problema de Diffie-Hellman (búsqueda):*

Sea $G$ un grupo y $n$ un número natural. Dados un elemento $g \in G$ de orden $n$ y dos elementos $g^a$ y $g^b$ donde $a, b \in \{1, \ldots, n-1\}$, encontrar el elemento $g^{ab}$.

*Problema de Diffie-Hellman (decisión):*

Sea $G$ un grupo no abeliano y sean dos elementos $g, c \in G$. Dados dos elementos $g^a$ y $g^b$ donde $a, b \in \{1, \ldots, ord(g)-1\}$, decidir si el elemento $c$ es igual a $g^{ab}$.

Sea $\mathbb{F}_p^* = (\mathbb{Z}/p\mathbb{Z})^* = \{1, 2, \ldots, p-1\}$ el grupo multiplicativo de los enteros módulo $p$ siendo $p$ un número primo. Sea $g \in \mathbb{F}_p^*$ un elemento primitivo módulo $p$, que consideraremos nuestra "base".

Supongamos que Alice y Bob quieren intercambiar una clave secreta para comunicarse. Para ello se procede de la siguiente forma:

1. Alice y Bob acuerdan elegir un grupo cíclico finito $G$ y un elemento generador $g$ de $G$. Escribimos el grupo multiplicativo $G$.
2. Alice elige un número natural $a$ aleatorio tal que $a \in \{1, \ldots, p-1\}$ y envía el elemento $g^a$ a Bob.
3. Bob elige un número natural $b$ aleatorio también tal que $b \in \{1, \ldots, p-1\}$ y envía el elemento $g^b$ a Alice.
4. Alice calcula $K_A = (g^b)^a = g^{ba}$.
5. Bob calcula $K_B = (g^a)^b = g^{ab}$.

Notar que como el grupo cíclico $\mathbb{Z}$ es conmutativo, entonces $ab = ba$, lo que implica que tanto Bob como Alice obtienen el mismo elemento del grupo $K = K_A = K_B$. El elemento $K$ será lo que utilicen como clave secreta.

El protocolo se considera seguro si tanto el grupo $G$ como el elemento generador $g$ se eligen correctamente. Tener en cuenta que el espía ha de resolver el problema de *Diffie-Hellman* para obtener la clave secreta. Más concretamente, deberá averiguar cuál es el elemento $g^{ab}$ a partir de los elementos $g^a$ y $g^b$.

Por otro lado, cabe destacar que el problema de Diffie-Hellman se asemeja al problema del logaritmo discreto, aunque no se consideran equivalentes.

*Problema del logaritmo discreto:*

Sea $G$ un grupo, $g \in G$ un elemento de orden $n$. Dado un elemento $y \in \langle g \rangle$, encontrar un elemento $x$ tal que $y = g^x$.

En efecto, si encontramos un algoritmo capaz de resolver el problema del logaritmo discreto para un grupo $G$, también se podrá resolver el problema de Diffie-Hellman, ya que si el algoritmo encuentra el valor de la potencia $x = ab$ entonces, como el elemento $g$ es conocido, se podría calcular $g^{ab}$ (lo cual resuelve el problema de la búsqueda). En otro caso, si $c$ es un elemento conocido, se podría verificar si $c = g^{ab}$ (lo cual resuelve el problema de decisión).

Notar que en caso contrario, se puede intentar resolver por "fuerza bruta", para lo cual bastaría calcular la potencia empezando desde el número 1 y comprobando si coincide o no el valor. Como el objetivo sería encontrar una potencia de $g$, el coste computacional será como máximo $\mathcal{O}(|g|)$ siendo $|g|$ el orden de $g$, lo que equivale a probar con todas las potencias del grupo cíclico. Por lo general, este orden es de gran magnitud, luego recurrir a la "fuerza bruta" no es un método nada eficaz.

El intercambio de claves de Diffie-Hellman fue utilizado, por otro lado, para la construcción de un famoso criptosistema de clave pública, ElGamal. Se describe de la siguiente manera:

1. Al igual que en Diffie-Hellman, Alice y Bob se ponen de acuerdo en la elección de un grupo cíclico finito $G$ de orden $p$ y de un elemento $g \in G$ que genere $G$.
2. Bob elige un elemento $b \in \{1, \ldots, p-1\}$ que será su clave privada, y calcula $c = g^b$.
3. La clave pública será entonces la tupla $(p, g, c)$.

Supongamos que Alice quiere enviar un mensaje $m$ a Bob, entonces

- Encriptar: Alice elige un elemento $a \in \{1, \ldots, p-1\}$ y calcula el elemento $g^a$. A continuación, el mensaje $m \in G$ lo multiplica por $c^b = g^{ba}$. Alice envía a Bob la tupla $(m \cdot c^a, g^a)$.
- Desencriptar: Bob recibe la tupla $(m \cdot c^a, g^a) \in G \times G$, y usa su clave privada $b$ para recuperar el mensaje, ya que $ab = ba$

$$(m \cdot c^a) \cdot ((g^a)^b)^{-1} = m \cdot g^{ba} \cdot g^{-ab} = m.$$

A continuación, describiremos algunos de los problemas mencionados al principio.

## 3.2. Problema de la palabra

En 1911 Max Dehn estudiaba los grupos fundamentales de variedades cuando descubrió un problema especial, conocido actualmente como el problema de la palabra, y que consideró lo suficientemente importante como para ser objeto de estudio junto al problema del conjugado y al problema de los isomorfismos de grupos. A lo largo del tiempo se han intentado buscar algoritmos capaces de resolver el problema de la palabra en diferentes grupos. En los años 60, Pyotr Novikov y William Boon demostraron que existían grupos finitamente presentados en

los que el problema de la palabra no tenía solución, es decir, que es imposible construir un algoritmo válido. A partir de ahí, se propusieron algunos criptosistemas basados en el problema de la palabra en grupos finitamente presentados.

*Problema de la palabra (búsqueda):*

Dada una representación recursiva $\langle X \mid R \rangle$ del grupo $G$, y un elemento $g = 1$ de $G$ encontrar una presentación del elemento $g$ como producto de los generadores y de sus inversos a través de las relaciones de $R$.

*Problema de la palabra (decisión):*

Dado una presentación recursiva del grupo $G$ y un elemento $g \in G$, decidir si $g = 1$ o no en $G$.

*Problema de la elección de la palabra (decisión):*

Dada una presentación recursiva del grupo $G$ y los elementos elementos $g, w_1, w_2 \in G$, decidir si $g = w_1$ o $g = w_2$.

**Corolario 3.1.** *Si $F_N$ es el grupo libre en los generadores $x_1, \ldots, x_N$, entonces el problema de la palabra tiene solución.*

*Demostración.* Sea $w$ la palabra dada. Resolver el problema de la palabra consiste en verificar si $w$ define al elemento neutro en $F_N$, para ello basta reducir $w$ usando un proceso de reducción $\rho$ (ver proposición 1.1). Entonces $w$ define la identidad de $F_N$ si y solo si $\rho(w)$ es la palabra vacía. □

Concretamente, este problema tiene solución en $B_N$. Autores como D. Epstein y W. Thurston desarrollaron algoritmos que resolvían el problema en tiempo $\mathcal{O}(l(w)^2 N \log N)$ donde se utiliza la forma normal ponderada a la izquierda (ver secciones 2.2 y 2.3), estudio que mejoró posteriormente Birman, Ko, Lee, en [4], con su algoritmo de la forma canónica cuyo coste computacional es $\mathcal{O}(l(w)^2 N)$ donde $l(w)$ es la longitud de la palabra.

A continuación presentamos dos protocolos motivados por el problema de la palabra:

### 3.2.1. Esquema de Shpilrain-Zapata

El problema de la palabra ha sido uno de los problemas que se ha intentado utilizar en varios protocolos. Sin embargo, no siempre se ha obtenido un resultado exitoso. En [25] los

autores creen que se debe, entre otras razones, a que el problema que se planteaba no era exactamente el problema de la palabra sino el *problema de la elección de la palabra*. Shpilrain y Zapata decidieron plantear un protocolo que, aunque se basa en el problema de la palabra, tenía una ventaja: la existencia de grupos $G$ para los cuales existe un algoritmo que resuelve el problema de la palabra pero en un tiempo de ejecución exponencial. Los autores deciden aplicar el esquema a los grupos de pequeñas cancelaciones, sin embargo, veremos como que su planteamiento se puede aplicar también en el grupo de trenzas.

El protocolo se basa, en términos generales, en encontrar presentaciones isomorfas de dicho grupo, hasta obtener una presentación con relaciones de longitud 3 o 4. Esta última presentación se hará pública, de manera que el usuario que quiera encriptar el mensaje en codificación binaria pueda escoger palabras que son igual a 1 en la presentación pública (lo que sustituye al bit 1) o distinta a 1 (lo que sustituye al bit 0). Para entender al completo el protocolo, necesitamos responder las siguientes preguntas:

1. Dada una presentación $\Gamma$ de un grupo $G$

   *a*) ¿Cómo conseguir una presentación isomorfa $\Gamma'$ de $G$?

   *b*) Si $\Gamma \cong \Gamma'$ y algunas de las relaciones de $\Gamma'$ tienen longitud mayor de 4, ¿cómo se obtiene una presentación $\hat{\Gamma} \cong \Gamma'$ con las relaciones de longitud 3 o 4?

2. Si se quiere encriptar un mensaje escrito como una secuencia de bits de 0 y 1 ¿cómo escogemos una palabra $w$ tal que $w = 1$ o $w \neq 1$?

A continuación, se describe el método propuesto por los autores [25], que responde a las preguntas anteriores. Posteriormente lo aplicaremos al grupo de trenzas.

**1. Transformaciones de Tietze**

En teoría de grupos, las transformaciones de Tietze sirven de herramienta para transformar una presentación de un grupo $G$ en otra presentación (quizá más sencilla) de $G$. Estas transformaciones se aplican a los generadores del grupo y al conjunto de relaciones y son isomorfismos que se pueden invertir fácilmente. Se definen cuatro tipos de transformaciones:

- **(T1)** Añadir un generador junto con una relación: Dada una presentación se puede añadir un generador expresado como una palabra en función de los generadores originales. Por ejemplo, incluímos un nuevo generador que depende de $x$:

$$G = \langle x \mid x^3 = 1 \rangle \simeq \langle x, y \mid x^3 = 1,\ y = x^2 \rangle.$$

- **(T2)** Eliminar un generador: Si hay una relación donde un generador es igual a una palabra formada por otros generadores, entonces ese generador se puede eliminar y en todas las relaciones en las que aparece ser sustituido por la palabra equivalente. Por ejemplo, se puede eliminar $x$ en el siguiente caso:

$$G = \langle x, y, z \mid x = yz, y^2 = 1, z^2 = 1, x = x^{-1} \rangle \simeq \langle y, z \mid y^2 = 1,\ z^2 = 1,\ (yz) = (yz)^{-1} \rangle.$$

- **(T3)** Aplicar automorfismo: Se puede aplicar un automorfismo del grupo libre generado por $X$ a la presentación. Para eso, se aplica el automorfismo a las relaciones de $R$ y se sustituye $R$ por el conjunto obtenido. Por ejemplo

$$G = \langle x, y, z \mid x^3 = 1,\ x = yz \rangle \simeq \langle x, y, z \mid x^3 = 1,\ x = (xy)z \rangle.$$

donde se aplica el automorfismo $y \to xy$ , $x \to x$, $z \to z$.

- **(T4)** Cambiar relaciones: Si hay relaciones $r_{m+1}, r_{m+2}, \ldots$ que derivan de las palabras $r_1, \ldots, r_m$ entonces se pueden añadir al conjunto $R$. Y del mismo modo, si hay relaciones $r_{m+1}, r_{m+2}, \ldots$ que se deducen de otras palabras $r_1, \ldots, r_m$ se pueden eliminar del conjunto $R$. Por ejemplo, podemos eliminar la segunda relación ya que se deduce de la primera:

$$G = \langle x, y, z \mid x^3 = 1,\ x^6 = 1,\ x = yz \rangle \simeq \langle x, y, z \mid x^3 = 1,\ \ x = yz \rangle.$$

Así, dada un presentación $\Gamma$ y aplicando reiteradamente transformaciones de uno de os cuatro tipos, se obtiene una presentación isomorfa. Esta afirmación se demuestra con el teorema siguiente:

**Teorema 3.2.** (Ver [16], teorema 2, p.54). *Dadas dos presentaciones finitas $\langle X|R\rangle$, $\langle Y|S\rangle$ para un grupo $G$, se puede transformar una presentación en la otra mediante una secuencia de transformaciones de Tietze.*

A continuación, supongamos que alguna de las relaciones de la presentación obtenida $\Gamma'$ tiene palabras de más de cuatro letras, entonces utilizando de nuevo las transformaciones de Tietze, podemos responder a la pregunta $1.b)$. En particular, aplicando transformaciones de tipos $(T_1), (T_3)$ se consigue reducir una palabra en al menos una letra y aplicándolo reiteradamente, llegamos a una palabra de longitud 3 o 4. Más explícitamente,

- **(T1)** Si tenemos una palabra $r_1 = x_i x_j u$ con $1 \leq i, j \leq k$, podemos introducir un nuevo generador $x_{k+1}$ y una nueva relación $r_{m+1} = x_{k+1}^{-1} x_i x_j$ (conteniendo una palabra de la palabra $r_1$). Así, la nueva palabra que expresa la relación $r_1$ tendrá longitud menor que la original.
- **(T3)** Aplicar automorfismos de la forma $x_i \to x_i x_j^{\pm 1}$ o $x_i \to x_j^{\pm 1} x_i$ para $i \neq j$, $1 \leq i, j \leq k$, así se consigue reducir la longitud de una palabra en al menos un dígito.

Si aplicamos ambas transformaciones a la presentación $\Gamma'$ entonces la presentación $\hat{\Gamma}$ obtenida también será isomorfa a $\Gamma$ por el teorema 3.2. Además, si una palabra es igual a 1 en $\hat{\Gamma}$ entonces también lo será en $\Gamma'$.

El motivo por el que los autores proponen esta longitud de las palabras en la presentación $\hat{\Gamma}$ (la cual será pública) se debe a que plantean la resolución del problema de la palabra mediante el algoritmo de Dehn. Para una palabra $w$, dicho algoritmo se basa en buscar subpalabras que formen parte de relaciones, en especial estas subpalabras $u$ verifican que $r = uv$ con $r$ una relación y $v$ una subpalabra de longitud menor que la de $u$. De este modo, se sustituye $u$ por $v^{-1}$ reduciendo así la palabra $w$. En caso de que no exista ninguna $u$, $w \neq 1$. Si el lector está interesado en conocer y profundizar en este algoritmo, puede encontrar información en la siguiente web: http://www.unige.ch/math/folks/arjantse/Abs/dehn.ps.

**3. Generación de elementos**

Se explica a continuación brevemente la idea propuesta por los autores para generar tanto el elemento igual a 1 como el elemento distinto de 1. Una descripción más detallada se encuentra en [25].

- Para generar elementos iguales a 1 en la presentación:

  1. Se eligen $p$ palabras del conjunto de relaciones, sus inversos o sus conjugados mediante uno o dos letras, y se hace el producto obteniendo así una palabra $p_1$.
  2. A continuación se introducen de manera aleatoria en $p_1$, subpalabras de la forma $\sigma_i\sigma_j^{-1}$ obteniendo $p_2$.
  3. Posteriormente, sobre la palabra $p_2$ de izquierda a derecha se irán revisando las subpalabras, de manera que si se encuentra una subpalabra contenida en una relación, esa subpalabra se sustituya por su inverso. Y se continúa hasta llegar al final de la palabra.
  4. Finalmente, si existe alguna subpalabra de la forma $x_ix_i^{-1}$ o $x_i^{-1}x_i$ se cancela.

  El proceso se reitera $p$ veces, volviendo a incluir subpalabras de la forma $\sigma_i\sigma_j^{-1}$ y procediendo como se ha descrito hasta obtener una palabra $u$.

  El objetivo es conseguir una palabra $u$ que no se distinga de $w' \neq 1$, para lo que proponen añadir un generador a la izquierda de la palabra $u$, obteniendo la palabra $v = x_iu$ y reiterar los pasos del 2. al 4. al menos $l(v)/2$ veces.

- Para generar elementos distintos a 1:

  1. Elegir una palabra aleatoria $v$. Se pueden añadir más términos hasta ajustar los exponentes de cada letra de modo que la suma de los exponentes para cada generador sea cero y obtener $u$ con la misma longitud que la palabra $u$ descrita en el punto anterior.
  2. A continuación, se añade un generador en el lado izquierdo de $u$ y se aplican los pasos 2. - 4. al igual que en la generación del elemento igual a 1.

  Sin duda, esta elección aleatoria incita a preguntarse si realmente se está escogiendo una palabra distinta de 1. Por ello, en [25] los autores demuestran brevemente que la probabilidad de elegir $w \neq 1$ de forma "aleatoria" es muy próxima a 1, luego el método propuesto se puede considerar válido.

**Criptosistema de clave pública**

Sea $G$ un grupo con una presentación en el cual el problema de la palabra tiene solución y que es elegido de forma común entre los dos usuarios. Consideraremos en nuestro caso $G = B_N$.

- **Clave pública:** La presentación $\hat{\Gamma} = \langle \hat{X} | \hat{R} \rangle$ isomorfa a $\Gamma' = \langle X' | R' \rangle$ que cumple que todas las palabras del conjunto de relaciones $\hat{R}$ son de longitud 3 o 4. $\hat{\Gamma}$ se obtiene aplicando las transformaciones de Tietze a la presentación $\Gamma'$ como se ha explicado antes.
- **Clave privada:** El isomorfismo entre la presentación $\Gamma$ y $\Gamma'$. Notar que la presentación $\Gamma$ que se ha elegido del grupo $G$ también es privada, lo cual tiene sentido ya que el usuario que quiera encriptar un mensaje no necesita conocer $\Gamma$.

Suponer que Bob quiere enviarle un mensaje $m \in \{0,1\}^n$ ($n \in \mathbb{N}$) a Alice:

- **Encriptar:** Bob elige el mensaje $m \in \{0,1\}^n$. A continuación elige una palabra $w$ que es igual a 1 en $\hat{\Gamma}$ y una palabra $w'$ que es distinta de 1 en $\hat{\Gamma}$ siguiendo los pasos descrito arriba. Bob sustituye el bit 1 por la palabra $w$ y el bit 0 por la palabra $w'$. El mensaje será una palabra expresada en $w$, $w' \in \hat{\Gamma}$(o equivalentemente en $\Gamma$). Bob envía el mensaje a Alice.
- **Desencriptar:** Alice recibe el mensaje encriptado de Bob. Como se verifica $w = 1$, $w' \neq 1$ en $\Gamma'$, Alice podrá aplicar en primer lugar el isomorfismo de $\Gamma' \to \Gamma$. Así, obtiene los elementos correspondientes en $\Gamma$ y a continuación resuelve el problema de la palabra en $\Gamma$, lo que le permitirá identificar si una palabra es igual a 1 o distinta de 1 y así obtener el mensaje $m$ que le envía Bob.

### 3.2.2. Algoritmo de Wagner y Magyarik

Wagner y Magyarik proponen en [29] un criptosistema, aunque se podría decir más bien que sugieren una construcción general ya que incluso los autores en su artículo afirman que no está demostrado que sea un método seguro.

La idea es elegir un grupo $G = \langle X|R\rangle$ para el cual no exista un algoritmo en tiempo polinomial que resuelva el problema de la palabra, mientras que sí se pueda en un grupo cociente $G^* = G/N$ que se construye a partir del propio grupo $G$.
Supongamos que existen más relaciones

$$S = \langle s_1 = e, s_2 = e, \ldots, s_p = e\rangle$$

con las que podemos ampliar el conjunto $R$, de modo que si $N$ es el subgrupo normal de $G$ generado por las palabras $s_i$ con $1 \leq i \leq p$, entonces $G^*$ es el grupo cociente $G^* = G/N$. La aplicación $\phi : G \to G^*$ queda bien definida por ser la aplicación cociente de forma que dado un elemento $x \in G$ representado por una palabra $w$, $\phi(x)$ es justamente el elemento de $G^*$ representado por $w$. Además, es importante tener en cuenta la siguiente proposición:

**Proposición 3.3.** *Dadas dos palabras $w_1$, $w_2$, la función $\phi$ cumple dos propiedades:*

- *Si $w_1, w_2$ son equivalentes en $G$, entonces $\phi(w_1)$, $\phi(w_2)$ son equivalentes en $G^*$.*
- *Si $\phi(w_1), \phi(w_2)$ no son equivalentes en $G^*$, entonces $w_1$, $w_2$ no son equivalentes en $G$.*

Este resultado se tendrá en cuenta a la hora de elegir la clave pública, ya que el esquema exige que dado un elemento de $G^*$, se tendrá que resolver el problema de la palabra para saber a qué elemento de $G$ corresponde. Sin embargo, notar que este problema no es propiamente el problema de la palabra, sino el problema de elección de la palabra, que como aseguraban Shilprain y Zapata en [25] era uno de los motivos por los que el criptosistema no siempre funcionaba, quizá de ahí que los autores no lo consideren un método seguro.

#### Criptosistema de clave pública

En términos generales para el criptosistema se necesitarán más de dos palabras. Considerar un alfabeto o conjunto de palabras $W_G = \{ w_1, \ldots, w_k\}$ en $G$ de modo que no sean equivalentes

dos a dos en $G^*$ (por la proposición sabemos entonces que tampoco lo serán en $G^*$. Más concretamente, se define $W_G$ lo suficientemente grande de modo que a cada símbolo (ya sea letra, número, ...) del mensaje que se quiera enviar se le asocie una palabra $w_i$ para $i = 1, \ldots, k$.

- **Clave pública:** Formada por la presentación del grupo $G = \langle X|R\rangle$ y el conjunto de palabras $W_G$ en $G$.
- **Clave privada:** Formada por el conjunto de relaciones $S$ que se añade al conjunto $R$ para formar la presentación $\langle X|R \cup S\rangle$ del grupo cociente $G^*$, obteniendo así $G^*$ un grupo donde el problema de la palabra tiene solución.

Suponer que Alice quiere comunicarse con Bob y enviarle el mensaje $m$ que será una secuencia de símbolos:

- **Encriptar:** Alicia reemplaza cada símbolo por la palabra $w_i \in W_G$ asociada, obteniendo así una palabra $Y$ formada por la concatenación de los $w_i$ en el puesto correspondiente. Posteriormente, aplicando a $Y$ las relaciones de $R$ consigue una palabra equivalente $\tilde{Y}$ y la envía a Bob.
- **Desencriptar:** Bob recibe $\tilde{Y}$, que es un elemento también de $G^*$. Como en $G^*$ el problema de elección de la palabra tiene solución, aplica el algoritmo en $G^*$ para saber a qué palabras $w_i$ es equivalente $\phi(\tilde{Y})$ y así recuperar el mensaje.

Para construir este algoritmo, los autores proponen tres formas para elegir las relaciones extras $\{s_i = e\}$ de manera que las relaciones del conjunto $R$ sigan siendo triviales, es decir, $r_j = e$:

1. Eliminar un generador: $x_i = e$ para algún $i$.
2. Fusionar dos generadores en uno: $x_i x_j^{-1} = e$ o $x_i x_j = e$ para algún $i, j$.
3. Conmutar dos generadores: $x_i x_j x_i^{-1} x_j^{-1} = e$ o equivalentemente, $x_i x_j = x_j x_i$ para algún $i, j$.

**Ejemplo 6.** Se sabe que el problema de la palabra tiene solución en el grupo de trenzas, luego se puede pensar en una aplicación de este protocolo al grupo de trenzas considerando $G^* = B_N = \langle X^*|R^*\rangle$. De este modo, podríamos considerar un grupo $G = \langle X|R\rangle$ cuyo conjunto de relaciones $R$ fuese un subconjunto del conjunto de relaciones de $B_N$. Posteriormente, el conjunto $S$ estaría formado por las relaciones de $R^*$ que no están en $R$.

Para finalizar la sección recalcar que a pesar de describir solo estos dos protocolos existen otros, como el *esquema de Garzón y Zalcstein* en el cual aparecen grupos que no son finitamente presentados. Más información sobre este esquema podemos encontrarla en la sección 2.2.2. de [11].

## 3.3. Problema de la conjugación

Una de las posibles generalizaciones del problema del logaritmo discreto es el llamado *problema de la conjugación*. El problema de la conjugación tiene un gran interés ya que hay muchos problemas topológicos de especial relevancia que se basan en la conjugación ([23]).

Sin embargo, no existe un algoritmo polinomial que sea capaz de resolverlo en algunos grupos en particular (como en los grupos de trenzas) y es esta ventaja la que muchos científicos han aprovechado para destacar la seguridad de un protocolo.

Podemos diferenciar dos tipos de problemas:

*Problema de la conjugación (búsqueda):*

Dada una presentación recursiva de un grupo $G$ y dos elementos conjugados $g,h \in G$, encontrar un elemento en particular $x \in G$ tal que $h = xgx^{-1}$.

*Problema de la conjugación (decisión):*

Dada una presentación recursiva de un grupo $G$ y dos elementos $g,h \in G$, determinar cuando existe un elemento $x \in G$ tal que $h = xgx^{-1}$.

Antes de pasar a describir dos protocolos que usan los problemas que se acaban de describir, observamos que dado un algoritmo que resuelve el problema de la conjugación también resolverá el problema de la palabra.

**Proposición 3.4.** *Sea $G$ un grupo presentado de forma recursiva. Si el problema de conjugación se puede resolver en $G$ entonces el problema de la palabra también.*

*Demostración.* Notar que al conjugar un elemento $g \in G$ con el elemento identidad $1_G$, se sigue obteniendo el elemento $g$. Luego si en el problema de conjugación elegimos los elementos $g, 1_G \in G$ y este problema tiene solución en $G$ entonces también se verifica en particular para $x = 1_G$ y se tiene resuelto el problema de la palabra en $G$. □

A lo largo de los diferentes estudios se ha motivado la idea de generalizar, en cierto modo, el problema de conjugación. Algunos problemas resultantes son:

*Problema de la conjugación múltiple (búsqueda):*

Sea $s$ un número natural. Dadas dos $s-$tuplas de elementos de un grupo $G$, $(a_1,\ldots,a_s)$, $(b_1,\ldots,b_s)$, encontrar un elemento $x \in G$ tal que $b_i = xa_ix^{-1}$ $\forall i$.

*Problema de la conjugación generalizada (búsqueda):*

Sea $G$ un grupo no abeliano, y $H \leq G$. Sean dos elementos $g,h \in G$ que son conjugados por un elemento de $H$, encontrar $x \in H$ tal que $h = xgx^{-1}$.

La seguridad computacional de los dos protocolos que se presentan a continuación está basada en la dificultad de resolver los problemas de conjugación y las ecuaciones de los grupos elegidos. Sin embargo, se sabe que existen algunos grupos donde problemas como el problema de la palabra tiene solución (es más, se puede resolver en tiempo polinomial) pero el problema de conjugación no. Un ejemplo es el grupo de trenzas con $N$ hebras, en el cual dada una palabra $w$ se puede resolver el problema de la palabra en un tiempo computacional $O(|w|^2N)$, mientras que el problema de conjugación necesita al menos un tiempo de ejecución exponencial.
Resultados como el de Birman-Ko-Lee muestran protocolos construidos usando los grupos de trenzas, y la simplicidad de dichos métodos han llevado a potenciar cada vez más el uso de grupos de trenzas para la criptografía de clave pública ([4]).

### 3.3.1. Protocolo de Ko-Lee

Ko-Lee observaron que los grupos de trenzas eran una opción muy atractiva para establecer un protocolo de intercambio de claves. En su artículo [14] describen un nuevo criptosistema de clave pública.

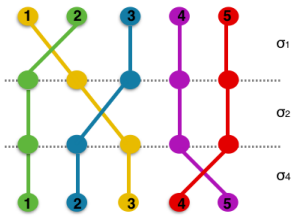
El grupo plataforma que usa es el grupo de trenzas $B_N$. Más concretamente, proponen escoger dos subgrupos $LB_l$ y $RB_r$ de $B_N$ con $N = l + r$. El subgrupo $LB_l := \langle \sigma_1, \dots \sigma_{l-1} \rangle$ está generado por los $l-1$ primeros generadores de Artin de $B_N$, es decir, y sus elementos son las trenzas que surgen del trenzado de las $l$ hebras de la izquierda. Análogamente, el subgrupo $RB_r := \langle \sigma_{l+1} \dots \sigma_{l+r+1} \rangle$ está generado por los $r-1$ últimos generadores de Artin de $B_N$, es decir, sus elementos son las trenzas que surgen del trenzado de las $r$ hebras de la derecha.

*Problema del protocolo de Ko-Lee (búsqueda):*

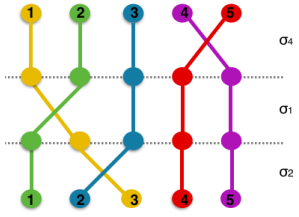
Sean $x \in B_N$, $a \in LB_l$ y $b \in RB_r$ y sean $y_1 = axa^{-1} \in B_N$ y $y_2 = bxb^{-1} \in B_N$. Dada la tupla $(x, y_1, y_2)$, encontrar el elemento $by_1b^{-1}$.

Notar que $abxa^{-1}b^{-1} = ay_2a^{-1} = by_1b^{-1}$. Esto es consecuencia de que los elementos de estos dos subgrupos $LB_l$, $RB_r$ conmutan. Esta propiedad se cumple siempre ya que $\sigma_l$ es el generador que geométricamente representa el cruce de la hebra $l$ (que está en $LB_l$) bajo la hebra $l+1$ (que pertenece a $RB_R$) y sin embargo, no está ni en el subgrupo $LB_l$ ni en $RB_r$ por lo que los movimientos que se produzcan en las $l$ primeras hebras no afectan a las $r$ últimas y viceversa. Eso implica la conmutatividad entre los subgrupos que será esencial para la definición del intercambio de claves, lo que dará como resultado la clave común entre Alice y Bob. En las siguientes figuras se muestra un ejemplo para $n = 5, l = 3, r = 2$, donde $A = \sigma_1\sigma_2 \in LB_l$ y $B = \sigma_4 \in RB_r$.

Si observamos el enunciado del problema del protocolo de Ko-Lee, se puede identificar la siguiente función de una vía, ya que el resultado de aplicar dicha función son dos elementos




$AB = \sigma_1\sigma_2\sigma_4$, permutación $\pi = 31254$




$BA = \sigma_4\sigma_1\sigma_2$, permutación $\pi = 31254$

conjugados de $x$, y aunque el cálculo del conjugado es sencillo, recuperar $a,b$ para obtener la $3-$tupla inicial es mucho más complejo.

$$\begin{aligned} f : LB_l \times RB_r \times B_N &\rightarrow B_N \times B_N \times B_N \\ (\, a,b,x\,) &\rightarrow (axa^{-1}, bxb^{-1}, x). \end{aligned} \tag{3.1}$$

Sin embargo, los autores propusieron otra función de una vía:

$$\begin{aligned} f : LB_l \times B_N &\rightarrow B_N \times B_N \\ (\, a\ ,\ x\,) &\rightarrow (axa^{-1}, x). \end{aligned} \tag{3.2}$$

Esta última función sabemos que presenta mucha dificultad ya que corresponde al problema de la conjugación generalizada. En este caso, se considera el grupo $G = B_N$ y $H = LB_l$. Sin embargo, no se puede asegurar que sean el mismo problema, ya que las funciones de una vía no son iguales. No obstante, se puede intuir una dificultad similar, pero para asegurarse los autores propusieron en [14] algunas propiedades para la trenza $x \in B_N$ con el objetivo de conseguir un problema más difícil:

1. Ya que cada trenza de $B_N$ puede escribirse de manera única usando la forma canónica, se propone reescribir las trenzas $x,a,b$ obteniendo su correspondiente forma normal a la izquierda. Para ello, se puede aplicar la teoría descrita en la sección 2.3.

2. Además, la trenza $x$ será parte de la información pública del criptosistema, por lo que se puede elegir como $x$ una trenza pura. Eso implica, por definición (2.1), que $\varphi(x) = Id_{S_N}$ siendo $\varphi : B_N \rightarrow S_N$ el epimorfismo canónico. Como consecuencia $\varphi(y_1) = \varphi(a)\varphi(x)\varphi(a^{-1})$ $= Id_{S_N}$ y análogamente, $\varphi(y_2) = Id_{S_N}$. Luego es extremadamente complicado recuperar $\varphi(a)$, $\varphi(b)$ y a partir de ahí, los valores de $a$ y $b$.

Teniendo en cuenta estas recomendaciones se describe el intercambio de claves y el criptosistema propuestos:

**Intercambio de claves**

Este esquema presenta una similitud con el intercambio de claves propuesto por Diffie-Hellman: un elemento del grupo como clave pública y elementos que conmutan como claves privadas.

Este algoritmo se basa previamente en una información común entre dos los usuarios. Sean $N,l,r$ números enteros tal que $l + r = N$. Se comienza con la elección de una trenza pura $x$ de $B_N$, dicha trenza se reescribirá y se publicará en su forma normal a la izquierda.

Suponer que Alice quiere intercambiar con Bob una clave secreta.

- Alice elige un trenza secreta $a \in LB_l$, la reescribe para obtener la palabra $a$ en su forma normal a la izquierda y le envía a Bob el elemento conjugado $y_1 = axa^{-1} \in B_N$.
- Bob elige la trenza secreta $b \in RB_r$, la reescribe obteniendo la palabra equivalente en su forma normal a la izquierda y envía a Alice el elemento conjugado $y_2 = bxb^{-1} \in B_N$.
- Alice recibe la trenza $y_2$ y de nuevo conjuga ese elemento con su clave privada $a$, es decir, $K_a = ay_2a^{-1} \in B_N$.
- Bob, análogamente, recibe $y_1$ y conjuga dicho elemento con su clave privada $b$, es decir, $K_b = by_1b^{-1} \in B_N$.

Debido a que $LB_l$ y $RB_r$ conmutan, se deduce que la trenza obtenida por Bob y por Alice es la misma (aunque pueden tener distinta representación)

$$ay_2a^{-1} = abxb^{-1}a^{-1} = baxa^{-1}b^{-1} = by_1b^{-1}$$

Por tanto, como $K_a = K_b$, este elemento se puede usar como clave secreta entre Alice y Bob. Es más, como el grupo tiene un algoritmo eficaz para computar la forma normal de los elementos del grupo, entonces la clave secreta $K$ podría ser la forma normal a la izquierda de $K_a$.

El intercambio de claves que acabamos de describir nos sirve para establecer un criptosistema de clave pública.

**Criptosistema de clave pública**

La información pública que se establece previamente es la siguiente: sean $N, l, r$ números naturales tal que $l + r = N$ y $LB_l$, $RB_r \leq B_N$. Sea $x \in B_N$ una trenza pura, de modo que la palabra que la representa $W_x$ es no reducible. El mensaje $m \in \{0,1\}^n$ que se quiere enviar está expresado en código binario, mientras que las claves (como se muestra a continuación) son trenzas. Por ello, se necesita una función *hash* (ver definición previa a 1.4) que permita expresar las trenzas que intervienen en el criptosistema como una secuencia binaria, es decir, una función *hash* $H : B_N \to \{0,1\}^n$ donde $\{0,1\}^n$ es el espacio de mensajes. En [14] no se dan más detalles sobre esta función, sin embargo, algunos estudios ([7]) proponen usar la forma norma ponderada a la izquierda.
Más concretamente, se sabe que toda trenza $\beta \in B_N$ (por el teorema 2.9) se puede expresar de manera única en su forma normal ponderada a la izquierda $W_\beta = \Delta^p A_1 \cdots A_k$. Esta expresión se puede traducir en una $(k+1)-$tupla donde $p$ es la potencia de la trenza fundamental $\Delta$ y $\pi_i$ son las permutaciones asociadas de las trenzas de permutación positivas $A_i$ para $i = 1, \ldots, k$ (descrito en la sección 2.2). Los elementos de esta tupla son números en código decimal (los denotamos con el subíndice 10) que se pueden transformar en código binario (lo denotamos con el subíndice 2). Así concatenando (en orden) las entradas de la tupla es código binario, se obtiene un elemento de $\{0,1\}^n$, donde $n$ es lo suficientemente grande.

Más formalmente, la función *hash* propuesta es la siguiente:

$$\begin{array}{ccccc} H : B_N & \longrightarrow & \mathbb{Z}^{k+1} & \longrightarrow & \{0,1\}^n \\ a = \Delta^p A_1 \cdots A_k & \mapsto & (p, \pi_1, \cdots, \pi_k)_{10} & \mapsto & (p, \pi_1, \cdots, \pi_k)_2. \end{array}$$

Teniendo en cuenta la información pública, podemos describir a continuación las claves que se usarán en el criptosistema.

- **Clave pública:** Formada por el par $(x,y)$ donde $y = axa^{-1}$ es un elemento conjugado de $x$ y $a \in LB_l$ será el elemento privado.
- **Clave privada:** Se considerará una trenza $a \in LB_l$, expresada en su forma normal a izquierda, y será el elemento que intervenga en la clave pública.

Suponer que Alice quiere mandarle el mensaje $m \in \{0,1\}^n$ a Bob. Ambos conocen la clave pública y la clave privada, entonces:

- **Encriptar:** Alice elegirá de forma aleatoria un elemento $b \in RB_r$ y computa el conjugado $c = bxb^{-1} \in B_N$. El elemento obtenido es una trenza a la que aplicará la función hash para transformala en una secuencia de 0 y 1 (para lo cual deberá expresarla en su forma normal a la izquierda). El mensaje está codificado también con 0 y 1, luego se podrá calcular $d = H(byb^{-1}) \oplus m \in \{0,1\}^n$ donde $\oplus$ representa la suma bit a bit en módulo 2. Finalmente, Alice envía la tupla $(c,d)$ a Bob.
- **Desencriptar:** Bob recibe el mensaje $(c,d)$. Como conoce la clave secreta, puede recuperar el mensaje sumando bit a bit (módulo 2)

$$\begin{aligned} H(aca^{-1}) \oplus d &= H(abxb^{-1}a^{-1}) \oplus (H(byb^{-1})) \oplus m \\ &= H(abxb^{-1}a^{-1}) \oplus (H(byb^{-1})) \oplus m \\ &= H(byb^{-1}) \oplus (H(byb^{-1})) \oplus m \\ &= m \end{aligned}$$

La seguridad de este protocolo, como ya se ha indicado anteriormente, se basa en la dificultad del problema de conjugación. Sin embargo, dependiendo de la situación, sería suficiente con resolver el problema de la descomposición [26].

> *Problema de descomposición (búsqueda):*
>
> Sean $w,x,y \in G$, encontrar elementos $x'$, $y' \in G$ tal que $x' \cdot w \cdot y' = x \cdot w \cdot y$

Suponer que un adversario conoce la trenza $x \in B_N$, $y = axa^{-1}$, y $c = bxb^{-1}$ y busca los elementos $a_1, a_2, b_1, b_2$ tal que :

- $a_1 x a_2 = y$.
- $b_1 x b_2 = c$.

de forma que $b_1, b_2 \in RB_r$ y $a_1, a_2 \in LB_l$ y así conmuten con el elemento $b \in RB_r$ (lo cual se cumple aunque no conozcan $b$ por la propiedad conmutativa de $RB_r$ y $LB_l$). Entonces podría computar

$$a_1 b_1 x b_2 a_2 = a_1 c a_2 = a_1 b^{-1} x b a_2 = b^{-1} a_1 x a_2 b = b^{-1} y b = b^{-1} a^{-1} x a b = K$$

Luego, si el adversario resuelve dos veces el problema de descomposición primero para $a_1, a_2$ y luego para $b_1, b_2$, y así recuperaría la clave privada. Notar, que este problema se considera más sencillo ya que consiste en resolver una ecuación con dos incógnitas, mientras que en el problema de la conjugación hay que resolverla teniendo una sola incógnita.

### 3.3.2. Protocolo de Anshel-Anshel-Goldfeld

En este trabajo hemos visto ya varios ejemplos de esquemas basados en grupos de trenzas. Esta línea de investigación fue iniciada por Anshel, Anshel y Goldfeld. El siguiente protocolo es un intercambio de claves que fue descrito en un primer momento en términos generales sin especificar un grupo plataforma. Fue más tarde, en [2], donde Anshel, Anshel, Fisher y Goldfeld presentaron este intercambio de claves para el caso particular del grupo de trenzas $B_N$ incluyendo también la representación coloreada de Burau de $B_N$ (ver sección 2.4). La seguridad de este protocolo se basa en el problema de conjugación múltiple y en una de sus principales características: la existencia de un extractor de claves. Un *extractor de claves* es una función, que denotamos $\mathscr{E}$, que asigna a cada uno de los elementos de un grupo $G$ una única clave. Los autores sugieren un extractor de claves de manera que a cada trenza $\beta \in B_N$ se le asigna su correspondiente representación coloreada de Burau $(CB(\beta), \alpha_\beta) \in GL_{N-1}(\mathbb{Z}[t_i^{\pm 1}]) \times S_N$ y posteriormente se evalúa la matriz $CB(\beta)$ en una lista de valores que se determinan previamente.

Más formalmente, se fija un número primo $p$. Sea $K_{N,p}$ el espacio de claves compuesto por las tuplas $(M, \alpha) \in GL_{N-1}(\mathbb{F}_p) \times S_N$. Sea $\{\tau_1, \ldots, \tau_n\}$ una lista de números enteros distintos e invertibles en el cuerpo $\mathbb{F}_p$. El extractor de claves se define como

$$\begin{array}{ccccc} \mathscr{E} : B_N & \longrightarrow & CB_N & \longrightarrow & K_{N,p} \\ \beta \mapsto (CB(\beta), \alpha_\beta) := (M_\beta(t_1, \ldots, t_N),\ \alpha_\beta) & \mapsto & (M_\beta(\tau_1, \ldots, \tau_N)\ \ mod p,\ \alpha_\beta) & & \end{array}$$

donde la reducción (mod $p$) se aplica para cada una de las entradas de la matriz $M_a(\tau_i)$.

**Intercambio de claves:**

Una vez introducido el extractor de claves, ya podemos describir el protocolo de Anshel-Anshel-Goldfeld que consiste en un intercambio de claves.

Se considera como información pública un número entero $N > 6$, un número $p$ primo tal que $p > N$, el extractor de claves y dos subgrupos de $B_N$

$$B_l = \langle a_1, a_2, \ldots, a_l \rangle$$

$$B_r = \langle b_1, b_2, \ldots, b_r \rangle$$

Suponer que Alice y Bob quieren compartir una clave, para ello cada uno elegirá su clave privada, publicará la clave pública correspondiente y a continuación podrá usar la clave pública del otro usuario y computar la clave compartida.
Explícitamente,

- Alice elige su clave secreta, que será una palabra $X$ escrita en términos de los generadores $a_i \in B_l$ para $i = 1, \ldots, N-1$ (es decir, las letras de la palabras son los $a_i$). A continuación la reescribe en su forma canónica (que será una palabra equivalente) y computa las claves públicas
$$X^{-1} b_1 X, X^{-1} b_2 X, \ldots, X^{-1} b_r X.$$

- Del mismo modo, Bob elige su clave secreta, que será una palabra $Y$ escrita en términos de los generadores $b_j \in B_r$ para $j = 1, \ldots, r$ (las letras de la palabra son $b_i$) y la reecribe

mediante su forma canónica en una palabra equivalente. Posteriormente, computa las claves públicas

$$Y^{-1}a_1Y, Y^{-1}a_2Y, \ldots, Y^{-1}a_lY.$$

- Para conseguir la clave compartida Alice computa $Y^{-1}XY$. Observemos que puede hacerlo ya que conoce la expresión de $X$ en términos de las $a_i$, luego simplemente sustituye cada $a_i$ por la correspondiente clave $Y^{-1}a_iY$ y obtiene $Y^{-1}XY$. Posteriormente, multiplica por $X^{-1}$ por la izquierda, obteniendo $Y^{-1}X^{-1}YX$.

- Sin embargo, Bob no sigue un proceso simétrico. De forma análoga a como hace Alice, Bob computa el elemento $X^{-1}YX$. A continuación, multiplica por $Y^{-1}$ por la izquierda, y computa el elemento inverso del resultado, obteniendo $(X^{-1}Y^{-1}XY)^{-1} = X^{-1}Y^{-1}XY$.

- Finalmente, tanto Alice como Bob aplican el extractor de claves al elemento $X^{-1}Y^{-1}XY$. La propia definición del extractor de claves nos asegura a ese elemento le asocia una clave, la cual será considera como la clave compartida de Alice y Bob:

$$\mathscr{E}(X^{-1}Y^{-1}XY) \equiv \mathscr{E}(K) = (\pi_K, M_K(\tau_1, \ldots, \tau_n) \ mod \ p).$$

Los autores no precisan que las palabras $X, Y$ de la clave privada tengan que ser reescrita. Sin embargo, podemos tener en cuenta la posibilidad de utilizar su forma canónica ya que el resultado es una palabra equivalente; en particular, la forma normal ponderada a la izquierda.

Notar además, que en [2] no describen un posible criptosistema en el que hacer uso de este intercambio de claves, como ocurre en el protocolo de Ko-Lee (sección 3.3.1), quizá se deba en parte a que los papeles de Alicie y Bob no son simétricos, lo cual presenta un inconveniente, ya que los intercambios de claves buscan que las partes involucradas estén en las mismas condiciones.
En el siguiente capítulo se estudia la firma WalnutDSA, un protocolo de firma digital también propuesto entre otros por Anshel-Anshel-Goldfeld, que usa una idea similar al extractor de claves que acabamos de describir.

**Nota.** A lo largo de todo el capítulo se propone que tras elegir una trenza (ya sea porque actúa como clave privada o como clave pública), dicha trenza se puede reescribir expresándola en su forma normal ponderada a la izquierda. Este hecho es una mera recomendación, ya que cualquier expresión de la trenza en una forma canónica sería válida, como por ejemplo, la forma normal de Birman ([4]).

Concluimos el capítulo con la descripción de dos problemas más considerados relevantes dentro de la teoría combinatoria de grupos:

*Problema de pertenencia (búsqueda):*

Dada un presentación recursiva del grupo $G$, un subgrupo $H \leq G$ generado por $h_1, \ldots, h_k$ y un elemento $g \in G$, decidir si $g \in H$ o no.

*Problema de pertenencia (decisión):*

Dada un presentación recursiva del grupo $G$, un subgrupo $H \leq G$ generado por $h_1, \ldots, h_k$ y un elemento $h \in H$, encontrar una expresión de $h$ en función de $h_1, \ldots, h_k$.

La resolución de este problema en algunos grupos resulta complicada. En particular, algunos matemáticos estudiaron el caso del grupo de trenzas $B_N$ para el cual no existe solución si $N \geq 6$. Esto se debe a que existen subgrupos isomorfos a $F_2 \times F_2$ (por ejemplo, el subgrupo generado por $\sigma_1^2$, $\sigma_2^2$, $\sigma_4^2$, $\sigma_5^2$) y para este grupo Mihailova probó que no hay un algoritmo que resuelva este problema ([28]).

Y por último, otro problema de decisión, que fue identificado por Max Dehn (1911) como uno de los tres problemas de decisión fundamentales en la teoría de grupos, al igual que el problema de la palabra y el problema de conjugación.

*Problema del isomorfismo:*

Dados dos grupos finitamente presentados $G_1$, $G_2$, decidir si son o no isomorfos.

# Capítulo 4

# Algoritmo WalnutDSA

En 1976, Diffie y Hellman describieron por primera vez la noción de un esquema de firma digital, pero fue con la invención del algoritmo RSA cuando se crearon por primera vez las firmas digitales primitivas. Más tarde, apareció el conocido algoritmo DSA (1991).

WalnutDSA es un esquema de firma digital propuesto por Anshel, Atkins, Goldfeld y Gunnels. Fue aceptado por el Instituto Nacional de Normas y Tecnología, con el objetivo de evaluar la resistencia cuántica de un criptosistema de clave pública. Entre sus características destaca el uso de una función de una vía, denominada *E-multiplicación*, que fue precisamente la que se consideró de alta resistencia cuántica, aumentando así el interés hacia este esquema de firma digital.

En este capítulo se estudia el protocolo correspondiente a la firma WalnutDSA. Para ello, se tendrán que cuenta algunas de las nociones ya introducidas en los capítulos 1, 2, 3. Su relevancia en este trabajo brota de la elección del grupo de trenzas como grupo plataforma, principal objeto de estudio. En este caso, la seguridad se basará en un nuevo problema diferente al problema de conjugación y al problema de la palabra. Previo a la descripción del algoritmo de la firma digital será necesario introducir el concepto de E-multiplicación y de elementos de "camuflaje" ([19], [3]).

## 4.1. E-multiplicación

La E-multiplicación es una función de una vía que se introdujo para crear múltiples construcciones algebraicas. Esta función combina las trenzas ($B_N$), las matrices ($GL_{N-1}(\mathbb{F}_q[t_i])$) y los cuerpos finitos ($\mathbb{F}_q$).

Recordar que $\mathbb{F}_q$ es el cuerpo finito de $q$ elementos y que $GL_{N-1}(\mathbb{F}_q[t_i^{\pm 1}])$ es el grupo de matrices invertibles sobre el anillo de los polinomios de Laurent en las variables $t_i$ para $i = 1,\ldots,N$ (definido en la sección 2.4).

Dada una lista de valores ordenados $\{\tau_1,\tau_2,\ldots,\tau_N\}$ de elementos no nulos de $\mathbb{F}_q$, podemos evaluar cada polinomio de Laurent en las variables $t_i$ sustituyendo la variable $t_i$ por $\tau_i$ para $i = 1,\ldots,N$ tal que:

$$\xi(f(t_1,t_2,\ldots,t_N)) := f(\tau_1,\tau_2,\ldots,\tau_N) \in \mathbb{F}_q$$

Esta operación se puede extender a las matrices $M(f_{ij})$ de $GL_{N-1}(\mathbb{F}_q[t_i^{\pm 1}])$ y define un homomorfismo de grupos

$$\begin{aligned} \xi : GL_{N-1}(\mathbb{F}_q[t_i^{\pm 1}]) &\longrightarrow GL_N(\mathbb{F}_q) \\ M(f_{ij}) &\longmapsto \xi(\, M(f_{ij})) := M(\xi(f_{ij})) \end{aligned}$$

**Definición.** Sea una matriz $M \in GL_N(\mathbb{F}_q[t_i^{\pm 1}])$. Sean las permutaciones $\alpha, \alpha_\beta \in S_N$ y una trenza $\beta \in B_N$. La operación *E-multiplicación* (denotada $\star$) se define como

$$(M, \alpha) \star (CB(\beta), \alpha_\beta) = (M', \alpha') \in GL_N(\mathbb{F}_q[t_i^{\pm 1}]) \times S_N$$

donde $(CB(\beta), \alpha_\beta)$ es la representación de Burau (definida en la sección 2.4) y $(M', \alpha')$ se define como sigue: si tenemos una de las trenzas $\beta = \sigma_i$ o $\beta = \sigma_i^{-1}$ para $i = 1, \ldots, N-1$, es decir, una trenza correspondiente a un único generador de Artin y $\alpha_i$ la permutación $(i \;\; i+1)$ correspondiente, entonces

$$(M, \alpha) \star (CB(\sigma_i^{\pm 1}),\ \alpha_i) = (M \cdot \xi({}^{\alpha}CB(\sigma_i^{\pm 1})),\ \alpha \cdot \alpha_i).$$

Y en el caso de tener una palabra de longitud mayor que uno, $\beta = \sigma_{i_1}^{\varepsilon_1} \sigma_{i_2}^{\varepsilon_2} \cdots \sigma_{i_k}^{\varepsilon_k}$ con $\varepsilon_i = \pm 1$, se define como

$$\begin{aligned} (M,\ \alpha) \star (CB(\beta),\ \alpha_\beta) &= (M \cdot \xi({}^{\alpha}CB(\sigma_i^{\pm 1})),\ \alpha \cdot \alpha_i) \\ &= (M,\ \alpha) \star (CB(\sigma_{i_1}^{\varepsilon_1}),\ \alpha_{i_1}) \star \cdots \star (CB(\sigma_{i_k}^{\varepsilon_k}), \alpha_{i_k}). \end{aligned}$$

Esta operación $\star$ define una acción de $CB_N$ en $GL_{N-1}(\mathbb{F}_q) \times S_N$. La operación E-multiplicación se entiende entonces como la acción de $B_N$ en $GL_{N-1}(\mathbb{F}_q) \times S_N$.

## 4.2. Elementos de "camuflaje"

La seguridad de WalnutDSA se debe a la existencia de algunas trenzas que pueden actuar como elementos de camuflaje.

**Definición.** Sea $v \in P_N$ una trenza pura de $B_N$. Fijamos un par $(M, \alpha)$ en $GL_{N-1}(\mathbb{F}_q[t_i^{\pm 1}]) \times S_N$. Se dice que $v$ es un *elemento de camuflaje* (cloaking element) de $(M, \alpha)$ si

$$(M, \alpha) \star (CB(v), \alpha_v) = (M, \alpha)$$

donde $M \in GL_{N-1}(\mathbb{F}_q[t_i^{\pm 1}])$ y $\alpha \in S_N$. Al conjunto de todos estos elementos de camuflaje se denota $Cloak_{(M,\alpha)}$.

Se puede demostrar que el conjunto $Cloak_{(M,\alpha)}$ es un subgrupo de $B_N$ y además, el elemento $v \in P_N$ estabiliza el elemento $(M, \alpha) \in GL_{N-1}(\mathbb{F}_q[t_i^{\pm 1}]) \times S_N$ usando la E-multiplicación. Recordamos que la definición de estabilizador de un elemento $x \in X$ (siendo $G$ un grupo que actúa sobre $X$) es

$$\mathbf{Stab}_G(x) = \{ g \in G \mid x = xg \,\}$$

Por lo que podemos deducir que, $Cloak_{(M,\alpha)} = \mathbf{Stab}_{P_N}(M, \alpha)$.

Por lo general, el problema de describir el estabilizador de un elemento suele ser complicado, de ahí que se hayan estudiado diferentes técnicas o algoritmos para construirlos. La siguiente proposición presenta un posible método que fue propuesto por los autores en [3].

**Proposición 4.1.** *Fijamos $N \geq 2$ y $1 < a < b < N$. Suponer que $\tau_a = \tau_b = 1$. Sean $M \in GL_{N-1}(\mathbb{F}_q[t_i^{\pm 1}])$ y $\alpha \in S_N$. Entonces, para cada $i = 1, \ldots, N-1$, $v = w\sigma_i^{\pm 2}w^{-1}$ es un elemento de camuflaje de $(M, \alpha)$ donde $\sigma_i$ es un generador de Artin y la permutación de $w$ cumple*

$$\alpha_w(i) = \alpha^{-1}(a) \quad \& \quad \alpha_w(i+1) = \alpha^{-1}(b).$$

Antes de proceder, introducimos la siguiente notación que facilitará la comprensión de la demostración. Recordemos la notación utilizada en la sección 2.4: para el generador de Artin $\sigma_i$ con $i = 1, \ldots, N-1$ con la permutación asociada $(i \;\; i+1)$ se define

$$CB(\sigma_i) = \left(\begin{array}{c|ccc|c} I_{i-2} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & t_i & -t_i & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & I_{N-i-1} \end{array}\right).$$

Entonces al aplicar la permutación $\alpha_i = (i \;\; i+1)$ sobre $CB(\sigma_i)$ (es decir, sobre el índice $i$ de $t_i$) nos queda

$$M_{\alpha_i}(\sigma_i) :=^{\alpha_i} CB(\sigma_i) = \left(\begin{array}{c|ccc|c} I_{i-2} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & t_{i+1} & -t_{i+1} & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & I_{N-i-1} \end{array}\right).$$

Denotemos ahora la matriz en cuya fila $i$ tiene

$$C_i(x) = \left(\begin{array}{c|ccc|c} I_{i-2} & 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & 0 & 0 \\ 0 & x & -x & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & I_{N-i-1} \end{array}\right).$$

Como la matriz $M_{\alpha_i}(\sigma_i)$ tiene la variable $t_{i+1}$, al sustituir el valor $t_{i+1}$ por el valor $\tau_{i+1}$ correspondiente nos queda

$$\xi(M_{\alpha_i}) = C_i(\tau_{i+1}).$$

Además, notar que $C_i(1)C_i(1) = I_N$. En efecto, basta multiplicar el bloque central de la matriz (ya que el resto son la identidad) tal que para $i = 2, \ldots, N-1$:

$$C_i(x) = \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Para el caso $i = 1$:

$$C_i(x) = \begin{pmatrix} -1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

*Demostración.* (Proposición 4.1) Basta demostrar que $(M, \alpha) \star (CB(v), \alpha_v) = (M, \alpha)$. En primer lugar, notar que :

$$(M, \alpha) \star (CB(v), \alpha_v) = (M, \alpha) \star (CB(w), \alpha_w) \star (CB(\sigma_i^{\pm 1}), \alpha_i) \star (CB(\sigma_i^{\pm 1}), \alpha_i) \star (CB(w^{-1}), \alpha_{w^{-1}}).$$

Para simplificar un poco la notación, denotamos $\sigma_i$ en lugar de $\sigma_i^{\pm 1}$ y recordamos que la permutación asociada al generador de Artin es la trasposición $(i \;\; i+1)$.
Se procede aplicando la E-multiplicación de izquierda a derecha, escogiendo los términos de dos en dos

$$(M,\alpha)\star(CB(w),\alpha_w) = (M\cdot\xi({}^{\alpha}CB(w)),\alpha\alpha_w).$$

Ahora, como $\alpha\alpha_w(i) = \alpha(\alpha^{-1}(a)) = a$,

$$\begin{aligned}(M\cdot\xi({}^{\alpha}CB(w)),\alpha\alpha_w)\star(CB(\sigma_i),\alpha_i) &= (M\cdot\xi({}^{\alpha}CB(w))\cdot\xi({}^{\alpha\alpha_w}CB(\sigma_i)),\alpha\alpha_w\alpha_i)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot\xi(M_{\alpha\alpha_w}(\sigma_i)),\alpha\alpha_w\alpha_i)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot C_i(\tau_a),\alpha\alpha_w\alpha_i)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot C_i(1),\alpha\alpha_w\alpha_i).\end{aligned}$$

A continuación, tener en cuenta que $(i,i+1)(i,i+1) = (i)(i+1)$ y que $\alpha\alpha_w(\alpha_i(i)) = \alpha(\alpha_w(i+1)) = b$. Entonces,

$$\begin{aligned}(M\cdot\xi({}^{\alpha}CB(w))\cdot C_i(1),\alpha\alpha_w\alpha_i)\star(CB(\sigma_i),\alpha_i) &= (M\cdot\xi({}^{\alpha}CB(w))\cdot C_i(1)\cdot\xi({}^{\alpha\alpha_w\alpha_i}CB(\sigma_i)),\alpha\alpha_w\alpha_i\alpha_i)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot C_i(1)\cdot\xi(M_{\alpha\alpha_w\alpha_i}(\sigma_i)),\alpha\alpha_w)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot C_i(1)\cdot C_i(\tau_b)),\alpha\alpha_w)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot C_i(1)\cdot C_i(1)),\alpha\alpha_w)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot I_N),\alpha\alpha_w).\end{aligned}$$

En la última igualdad se tiene en cuenta la observación que hay antes de la demostración: $CB(1)\cdot CB(1) = I_N$. Por último,

$$\begin{aligned}(M\cdot\xi({}^{\alpha}CB(w)),\alpha\alpha_w)\star(CB(w^{-1}),\alpha_{w^{-1}}) &= (M\cdot\xi({}^{\alpha}CB(w))\cdot\xi({}^{\alpha\alpha_w}CB(w^{-1}))),\alpha\alpha_w\alpha_{w^{-1}})\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot\xi({}^{\alpha}M_{\alpha_w}(w^{-1})),\;\;\alpha)\\ &= (M\cdot\xi({}^{\alpha}CB(w))\cdot\xi({}^{\alpha}CB(w^{-1})),\;\;\alpha)\\ &= (M\cdot\xi({}^{\alpha}CB(ww^{-1})),\;\;\alpha)\\ &= (M\cdot I_N,\;\;\alpha) = (M,\alpha).\end{aligned}$$

Queda demostrado que $v$ sí estabiliza al elemento $(M,\alpha)$.

□

## 4.3. Descripción del algoritmo

Suponer que Alice quiere enviar un mensaje a Bob, el siguiente algoritmo permitirá a Bob validar correctamente el mensaje de Alice a través de la clave pública. Para comprender el algoritmo se necesita una información inicial que será común para todos los usuarios:

Sea $B_N$ el grupo de trenzas de rango $N \geq 8$. Se establece una función o algoritmo $R : B_N \to B_N$ que permite reescribir una palabra de $B_N$ de modo que dada una palabra $w$ que representa una trenza de $B_N$, $R(w)$ es una palabra equivalente. Este algoritmo podría ser, por ejemplo, la forma normal a la izquierda descrita en la sección 2.3. Esta transformación permitirá en algunos casos ocultar información de la palabra original $w$, siendo así más complicado el ataque.
Por otro lado, sea el cuerpo finito $\mathbb{F}_q$ para $32 \leq q$ un número natural. Se consideran $N$ valores $\tau_1, \ldots, \tau_N$ no nulos de $\mathbb{F}_q$ entre los cuales dos de ellos tienen valor uno, $\tau_a = \tau_b = 1$ (con $1 \leq a < b \leq N$), lo cual permitirá aplicar la proposición 4.1. Sin embargo, esta condición no se exige siempre. En [19], los autores proponen elegir $\tau_b = -\tau_a^{-1}$ en lugar de $\tau_a = \tau_b = 1$ cuando los elementos de camuflaje sean de la forma $v = w\sigma_i^{\pm 4}w^{-1}$. Por lo que se deberá establecer qué tipo de elemento de camuflaje se van a usar.

Fijada esta información éstandard ya se puede describir la generación de claves y de la firma digital.

**Clave privada**

Sean $w, w'$ dos trenzas de $B_N$ representadas por palabras reducidas no triviales y que no son necesariamente trenzas puras. La clave privada para la firma $S$, $Priv(S)$, se determina como la tupla

$$Priv(S) = (w, w').$$

Esta clave se usa tanto para la clave pública como para la firma digital, por lo que cuanto mayor sea la longitud de las trenzas, más difícil será de recuperar dichas trenzas y con ello, atacar al esquema.

**Clave pública**

Dada una clave privada $(w, w')$, la clave pública de la firma $S$, $Pub(S)$, se define como

$$P(w) = \xi(CB(w), \alpha_w), \;\; P(w') = \xi(CB(w'), \alpha_{w'}).$$

Se determina la clave pública como la tupla

$$Pub(S) = (P(w), P(w')).$$

**Firma digital**

Como ya hemos comentado, la firma es una trenza que dependerá del mensaje $m$ que se envíe y por lo cual se necesita transformar el mensaje encriptado en una palabra que represente una trenza. El mensaje se escribe usando la codificación binaria de 0 y 1, de manera que $m \in \{0,1\}^*$. A continuación, se divide la secuencia en bloques de 4 dígitos $b_3b_2b_1b_0$, para lo cual se requiere una función *hash* $H : \{0,1\}^* \to \{0,1\}^{4k}$ obteniendo así $k$ bloques, donde $k$ es un número natural.
El método que proponen es asociar a cada uno de esos bloques una trenza, de modo que si el mensaje es la concatenación de los $k$ bloques, el mensaje encriptado será la concatenación de las $k$ trenzas. Se define así la función de encriptación $E : \{0,1\}^{4k} \to B_N$.

Al igual que ocurre en el protocolo de Ko-Lee (ver 3.3.1), el uso de trenzas puras permitirá que un adversario no pueda recuperar la clave a través de la permutación asociada. Se considera

entonces el siguiente conjunto de trenzas puras formado con los generadores de Artin $\sigma_i$:

$$\begin{aligned} g_{N-1,N} &= \sigma_{N-1}^2 \\ g_{N-2,N} &= \sigma_{N-1}\sigma_{N-2}^2\sigma_{N-1}^{-1} \\ g_{N-3,N} &= \sigma_{N-1}\sigma_{N-2}\sigma_{N-3}^2\sigma_{N-2}^{-1}\sigma_{N-1}^{-1} \\ &\vdots \\ g_{1,N} &= \sigma_{N-1}\cdots\sigma_2\sigma_1^2\sigma_2^{-1}\cdots\sigma_{N-1}^{-1} \end{aligned}$$

Recordamos que cada uno de los generadores corresponde a la transposición $(i\ \ i+1)$, y además $(i\ \ i+1)(i\ \ i+1) = Id_{S_N}$, por lo que la permutación asociada a cada elemento $g_{i,N}$ es $id_{S_N}$. Es decir, los elementos $g_{i,N}$ son trenzas puras y ninguna de ellas es un elemento trivial. Es más, si consideramos la concatenación o producto de varios de estos elementos, la palabra reducida obtenida es no trivial, consecuencia del elemento $\sigma_i^2$. De ello se puede deducir que el conjunto de trenzas definido forma una base de un subgrupo libre de $B_N$.
Por el teorema 1.5, simplemente escogiendo un subconjunto formado por cuatro de las trenzas definidas arriba, también se genera un subgrupo libre, al que denotamos $C_{N,4} = \{ g_{k_1,N}, g_{k_2,N}, g_{k_3,N}, g_{k_4,N}\}$. De este modo, podemos restringir $B_N$ al conjunto $C_{N,4}$ en nuestra función de encriptación y asignar a cada bloque del mensaje, una potencia de un elemento de $C_{N,4}$:

$$\begin{aligned} E : \{0,1\}^{4k} &\to C_{N,4} \\ b_3b_2b_1b_0 &\to g_{k_\mu}^{\upsilon} \end{aligned}$$

Originalmente lo autores no especificaron cómo calcular los valores de $\mu$, $\upsilon$, pero en [19] se propone calcular los valores como $\mu = 2b_1 + b_0 + 1$ y $\upsilon = 2b_3 + b_2 + 1$.
Entonces, para cada uno de los bloques obtenemos una potencia de $g_{k_\mu,N}$, y el mensaje encriptado $E(H(m))$ resulta ser la concatenación de todas las potencias $g_{k_\mu,N}$, y como hemos explicado antes, el resultado es una trenza pura de $C_{N,4}$.

**Ejemplo 7.** Sea $N = 5$ y consideramos el conjunto $C_{5,4} = \{ g_{1,5},\ g_{2,5},\ g_{3,5},\ g_{4,5}\}$ donde

$$g_{1,5} = \sigma_4\sigma_3\sigma_2\sigma_1^2\sigma_2^{-1}\sigma_3^{-1}\sigma_4^{-1} \qquad g_{2,5} = \sigma_4\sigma_3\sigma_2^2\sigma_3^{-1}\sigma_4^{-1}$$
$$g_{3,5} = \sigma_4\sigma_3^2\sigma_4^{-1} \qquad g_{4,5} = \sigma_4^2$$

Suponer que nuestro mensaje es *HOLA* y usamos el código ASCII para reescribirlo como una secuencia de 0 y 1:

| $H$ | $O$ | $L$ | $A$ |
|---|---|---|---|
| 01001000 | 01101111 | 01101100 | 01100001 |

Como cada una de las letras ya está codificada con 8 dígitos, basta separar cada letra en dos bloques y ver qué trenza le corresponde:

$$H \Rightarrow\ 0100 \to g_{1,5}^2\ \&\ 1000 \to g_{1,5}^3 \qquad O \Rightarrow\ 0110 \to g_{3,5}^2\ \&\ 1111 \to g_{4,5}^4$$
$$L \Rightarrow\ 0110 \to g_{2,5}^2\ \&\ 1100 \to g_{1,5}^4 \qquad A \Rightarrow\ 0110 \to g_{2,5}^2\ \&\ 0001 \to g_{2,5}$$

Así,

$$E(H(m)) = g_{1,5}^2 g_{1,5}^3 g_{3,5}^2 g_{4,5}^4 g_{2,5}^2 g_{1,5}^4 g_{2,5}^2 g_{2,5}$$

Una vez que hemos transformado el mensaje en una trenza, nos faltará incluir las claves privada y pública en la firma. Para ello, tengamos en cuenta la siguiente observación: la clave pública consiste en una tupla formada por dos representaciones de Burau, es decir, dos matrices y dos permutaciones. Sin embargo, para la firma necesitamos que sea una trenza (al igual que la clave privada y el mensaje encriptado). Aquí introducimos los elementos de camuflaje, ya que por definición, sabemos que para cada par de matriz - permutación podemos encontrar una trenza pura que actúa como elemento de camuflaje. Será esta trenza pura la que se use en lugar de la clave pública.

Siguiendo las explicaciones de 4.2, se buscan los elementos de camuflaje de $(Id_N, Id_{S_N})$, $P(w)$ y $P(w')$. Los denotamos respectivamente $v, v_1, v_2$, y por la proposición 4.2 estas trenzas de camuflaje se pueden elegir de forma que sus permutaciones inducidas son la identidad.

Una vez calculados $v, v_1, v_2$, ya tenemos todas las trenzas que necesitamos:

$$E(H(m)), w, w', v, v_1, v_2.$$

La firma $S$ que nos ayudará a verificar que ese es el mensaje esperado se obtiene de reescribir la trenza $v_1 \cdot w^{-1} \cdot v \cdot E(H(m)) \cdot w' \cdot v_2$ mediante la función pública $R$.

$$S = R(v_1 \cdot w^{-1} \cdot v \cdot E(H(m)) \cdot w' \cdot v_2)$$

Y así, finalmente, obtenemos que, si Alice quiere enviar el mensaje $m$, le enviará a Bob el mensaje cifrado y la firma para que pueda verificarlo, $(H(m), S)$.

Bob, que conoce las claves públicas de Alice, recibirá el mensaje $(H(m), S)$ y deberá comprobar que:

$$\mathscr{M}(P(w) \star S) = \mathscr{M}(P(E(H(m)))) \cdot \mathscr{M}(P(w'))$$

donde $\mathscr{M}(M, \alpha) = M$, es decir, indica la matriz de la tupla.

En definitiva, se dirá que la firma $S$ es válida si se cumple la igualdad anterior.

**Observación.** Recordamos el ejemplo 7, donde

$$E(H(m)) = g_{1,5}^2 g_{1,5}^3 g_{3,5}^2 g_{4,5}^4 g_{2,5}^2 g_{1,5}^4 g_{2,5}^2 g_{2,5}.$$

Si quisiésemos obtener una firma $S$ para este mensaje se debería elegir una clave privada lo suficientemente larga, así como un riguroso cálculo para obtener los elementos de camuflaje de la clave pública y de $(I_N, I_{S_N})$. Por lo que se puede intuir como con un simple ejemplo, el coste es elevado de forma manual.
Podemos encontrar un ejemplo detallado que describen los autores de la firma WalnutDSA en [3] (a partir de la p.22).

Concluímos el capítulo definiendo el problema base en el que se basa la seguridad de la firma WalnutDSA y el cual está descrito con mayor detalle en [3].

*Problema de la inversa de la E-multiplicación:*

Sea $N \geq 8$, $B_N$ el grupo de trenzas y $S_N$ el grupo simétrico. Sea $\mathbb{F}_q$ el cuerpo finito de $q$ elementos con $q \geq 32$ y se fija una lista de valores $\{\tau_1, \ldots, \tau_N\}$ en $\mathbb{F}_q$. Suponer que $\beta \in B_N$ es una trenza suficientemente larga escrita en su forma canónica y $(M, \alpha) \in GL_{N-1}(\mathbb{F}_q) \times S_N$ donde $P(\beta) = (M, \alpha)$. Determinar una trenza $\beta'$ tal que

$$(M, \alpha) = P(\beta').$$

# Capítulo 5

# Conclusiones

## 5.1. Futuras líneas de trabajo

A lo largo del trabajo se han expuesto diferentes protocolos aplicados a los grupos de trenzas. Algunos de los autores correspondientes estudian el esquema propuesto sin especificar cuál es el grupo plataforma o determinando un grupo plataforma distinto al grupo de trenzas como pueden ser: grupos de Artin, grupos de Grigorchuk, grupos de Thompson, grupos de matrices, grupos de pequeñas cancelaciones, etc. No obstante, de cara al futuro, el interés se centra en seguir la línea de investigación de este trabajo, con el fin de aprovechar los resultados expuestos sobre grupos de trenzas (y teoría de grupos en general), en grupos que tengan una estrecha relación con los grupos de trenzas, como pueden ser: los grupos de Artin y de Coxeter y los grupos de nudos [20], [13].

**Grupos de Artin y grupos de Coxeter**

En 1925, Artin estableció unos grupos que generalizaban de manera natural los grupos de trenzas, conocidos como los grupos de Artin. Podemos definir los grupos de Coxeter y de Artin como sigue.

Consideramos una matriz simétrica $M$ de tamaño $n \times n$, con entradas en $\mathbb{Z}_+ \cup \{\infty\}$ y con $m_{ii} = 2$. Este tipo de matrices se llaman *matrices de Coxeter*.

El *grupo de Artin* asociado a $M$ es el grupo dado por la presentación finita

$$\langle a_1, \ldots, a_n | R \rangle$$

donde el conjunto de relaciones $R$ se define como sigue: Sean $a_i, a_j$ dos de los generadores, si $i \neq j$ y $m_{ij} \neq \infty$ entonces en $R$ tenemos la relación:

$$a_i a_j \cdots = a_j a_i \cdots$$

donde ambas palabras tienen longitud precisamente $m_{ij}$. Si $m_{ij} = \infty$ o si $i = j$ entonces no hay relación entre $a_i$ y $a_j$ en $R$.

El *grupo de Coxeter* asociado a $M$ es el grupo dado por la presentación finita

$$\langle a_1, \ldots, a_n | \ R \rangle$$

donde el conjunto de relaciones $R$ es el siguiente: dados dos de los generadores $a_i, a_j$, si $m_{ij} \neq \infty$, entonces en $R$ tenemos la relación $(a_i a_j)^{m_{ij}} = 1$. Si $m_{ij} = \infty$ entonces no hay relación entre $a_i$ y $a_j$ en $R$.

En el caso del grupo de Coxeter, tenemos $a_i^2 = 1$ para cada $i$, pero en el caso del grupo de Artin los generadores tienen orden infinito. De hecho la presentación anterior del grupo de Coxeter se obtiene simplemente añadiendo las relaciones $a_i^2 = 1$ para $i = 1, \ldots, n$ a la presentación anterior del grupo de Artin de la misma matriz $M$.

Si tomamos como matriz $M$ una matriz con todas las entradas iguales a 2, excepto la diagonales inmediatamente superior e inmediatamente inferior a la principal, cuyas entradas son 3, el grupo de Artin asociado es un grupo de trenzas y el grupo de Coxeter asociado es el simétrico.

Por todo ello, como los grupos de trenzas son un caso particular de los grupos de Artin, se podría plantear la posibilidad de estudiar criptosistemas cuya plataforma sean los grupos de Coxeter o los grupos de Artin, utilizando la teoría algorítmica de estos grupos.

**Grupos de nudos**

Un *nudo* se define como una curva lisa, simple y cerrada, y se puede asociar a las trenzas cerradas [13]. Más concretamente, dado un diagrama de trenzas, los correspondientes extremos superiores de la hebras pueden conectarse con los inferiores.. Bastará darle una orientación a la trenza para obtener un nudo orientado. Se observa que dos trenzas iguales dan el mismo nudo, sin embargo, puede ocurrir que dos trenzas diferentes originen el mismo nudo. El problema se puede resolver gracias al teorema de Alexander [20]. A día de hoy existen investigaciones extensas sobre esta teoría y su relación con la criptografía. En [20] ya se plantea un criptosistema basado en el uso de nudos primos (nudos que no se pueden descomponer) donde Alicie y Bob deberán elegir una lista de nudos primos de manera secreta.

Actualmente es una rama que está en estudio y parece que tendrá una gran repercusión en la criptografía cuántica.

## 5.2. Conclusión

A lo largo del estudio nos hemos enfocado en la aplicación de los diferentes protocolos a los grupos de trenzas. Estos grupos destacan por sus buenas propiedades, entre ellas, la posibilidad de escribir cada una de las trenzas en su forma canónica: forma normal de Garside, la forma normal a la izquierda, la de Birman, etc. De ahí el interés de tener una función $R$ para reescribir una palabra que representa una trenza en una palabra equivalente es una característica que se puede aplicar en todos los protocolos y que permite conseguir una expresión única de cada elemento e incluso ocultar parte de la información. Esta función $R$ se puede considerar como una función de una vía, ya que recuperar la forma original de la trenza partiendo de la forma canónica es algo muy complejo.

El objetivo del trabajo ha sido presentar las diferentes herramientas y algoritmos que se han planteado y analizado sobre criptosistemas que usan la teoría de grupos, estudiando la parte matemática teórica que engloba a cada uno de los protocolos, así como los problemas base en los que se apoyan. Los tres principales problemas que hemos destacado son: el problema de la palabra, el problema de la conjugación y el problema de la inversión de la E-multiplicación, aunque no cabe duda de que hay y surgirán más.

Además, la dificultad de resolver el problema base se traduce en mayor seguridad, un objetivo fundamental en un algoritmo criptográfico.

En el mundo en el que vivimos, la información tiene un valor incalculable y su seguridad es un medio que hay que fortalecer, de ahí que la comunidad científica junto con los criptoanalistas estén continuamente estudiando y mejorando algoritmos. Vistos los artículos en los que se basa este trabajo, no cabe duda de que la criptografía basada en teoría de grupos jugará un papel muy importante. Además, como se menciona en el último capítulo, ya se está usando incluso para herramientas criptográficas con alta resistencia cuántica. Y es que, aunque la palabra *cuántico* suene lejana, los avances en esta rama no pasan desapercibidos y se prevée que el gran potencial de estos ordenadores u otros sistemas requieran de criptosistemas lo suficientemente complejos, un ámbito en el que todavía queda mucho por descubrir.

Como decía Alan Turing...

*Solo podemos ver poco del futuro,*
*pero lo suficiente para darnos cuenta*
*de que hay mucho que hacer.*

# Bibliografía


[1] IRIS ANSHEL, MICHAEL ANSHEL AND DORIAN GOLDFELD, *An algebraic method for public-key cryptography*.

[2] IRIS ANSHEL, MICHAEL ANSHEL, BENJI FISHER AND DORIAN GOLDFELD, *New Key Agreement Protocols in Braid Group Cryptography*. Topics in Cryptology CT-RSA 2001, vol.2020, Springer (2001), pp.13-27.

[3] IRIS ANSHEL, DEREK ATKINS, DORIAN GOLDFELD AND PAUL E.GUNNELLS, *Walnut*$^{TM}$*: A quantum-resistant digital signature algorithm*.
Disponible en web: https://veridify.com/wp-content/uploads/2017/12/walnutdsa.pdf

[4] JOAN BIRMAN, HI HYOUNG KO AND SANG JIN LEE, *A new approach to the word and conjugacy problems in the braid groups*. Advances in Math. 139 (1998), pp.322-353.

[5] SIMON R. BLACKBURN, *Recovering a private key in WalnutDSA*. Royal Holloway University of London, Egham, Surrey TW20 0EX, UK (2018).

[6] JAE CHOON CHA, KI HYOUNG KO, SANG JIN LEE, JAE WOO HAN AND JUNG HEE, *An efficient implementation of braid groups*. Department of Mathematics, Korea, pp.144-156.
Disponible en web: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.104.1830&rep=rep1&type=pdf

[7] PATRICK DEHORNOY, *Braid-based cryptography* (2004), pp.24.
Disponible en web: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.10.1759&rep=rep1&type=pdf

[8] ELSAYED A. ELRIFAI AND HUGH R. MORTON, *Algorithms for positive braids*. Quarterly Journal of Mathematics, version 2.3 (1991).
Disponible en web: https://www.researchgate.net/publication/31418546_Algorithms_for_positive_braids

[9] ————————, *Free groups*.
Disponible en web: https://www.math.unl.edu/~mbrittenham2/classwk/990s08/public/myasnikov.1.free.groups.pdf

[10] F. A. GARSIDE, *The braid group and other groups*. The Quarterly Journal of Mathematics (1967), pp.235-254 .

[11] Mª Isabel Glez Vasco, *Criptosistemas basados en Teoría de grupos*. Tesis doctoral, Departamento de matemáticas, Universidad de Oviedo (2003).
Disponible en web: http://digibuo.uniovi.es/dspace/bitstream/10651/16318/1/TD_MariaIsabel.pdf

[12] Juan González-Meneses, *Annales mathématiques Blaise Pascal*. Basic results on braid groups, vol.18, nº 1 (2011) pp.15-59.
Disponible en web: http://www.numdam.org/article/AMBP_2011__18_1_15_0.pdf

[13] Rebecca Hoberg, *Knots and braids*.
Disponible en web: http://www.math.uchicago.edu/~may/VIGRE/VIGRE2011/REUPapers/Hoberg.pdf

[14] Hi Hyoung Ko, Sang Jin Lee, Jung Hee Cheon, Jae Woo Han, Ju-sung Kang and Choonski Park, *New Public-key Cryptosystem using braid groups*. Advances in Cryptology - CRYPTO 2000, vol.1880, Springer, Berlin, Heidelberg, pp.166-183 .

[15] Sang Jim Lee and Eonkyung Lee, *Potencial Weaknesses of the Commutator Key Agreement Protocol based on Braid Groups*.
Disponible en web: https://iacr.org/archive/eurocrypt2002/23320013/LL02.ps

[16] D.L. Johnson, *Presentation of groups*. London Mathematical Society, Lecture Notes Series 22, Cambriadge University Press (1976), pp.52-60.

[17] Christian Kassel and Vladimir Turaev, *Braid groups*. num.3, Graduate Texts in Mathematics, Springer (2008), pp.85-96.
Disponible en web: https://biblioteca.unirioja.es/tfe_e/TFE001026.pdf

[18] Neal Koblitz, *Algebraic Aspects of Cryptography*. Algorithms and Computations in Mathematics, Springer-Verlag, Berlin, 3ª ed. (2004).

[19] Matvei Kotov, Anton Menshov and Alexander Ushakov, *An attack on the walnut digital signature algorithm*.
Disponible en web: https://eprint.iacr.org/2018/393.pdf

[20] Annalisa Marzuoli and Giandomenico Palumbo, *Post Quantum Cryptography from Mutant Prime*. International Journal of Geometric Methods in Modern Physics (2010).
Disponible en web: https://www.researchgate.net/publication/47374725_Post_Quantum_Cryptography_from_Mutant_Prime_Knots

[21] Alexei Myasnikov, Vladimir Shpilrain and Alexander Ushakov, *Group-based cryptography*. Montreal, Nueva York (Enero 2008).

[22] Alexei Myasnikov, Vladimir Shpilrain and Alexander Ushakov, *A practical attack on a braid group based cryptographic protocol*.

[23] Vladimir Shpilrain, *Problems in group theory motivated by cryptography*.
Disponible en web: https://www.researchgate.net/publication/323335090_Problems_in_group_theory_motivated_by_cryptography

[24] VLADIMIR SHPILRAIN AND GABRIEL ZAPATA, *Combinatorial group theory and public key cryptography*. Applicable Algebra in Engineering, Communication and Computing (2004), pp.291-302.

[25] VLADIMIR SHPILRAIN AND GABRIEL ZAPATA, *Using decision problems in public key cryptography*. Group-based Cryptography, pp.77-93.
Disponible en web: https://eprint.iacr.org/2007/147.pdf

[26] VLADIMIR SHPILRAIN AND ALEXANDER USHAKOV, *The conjugacy search problem in public key cryptography: unnecessary and insufficient*. Applicable Algebra in Engineering, Communication and Computing, Springer (2006), **17**, pp.285-289.

[27] B. SURY, *Free groups - basics*. Indian Statistical Institute, IIT Bombay, India (2010).
Disponible en web: https://www.isibang.ac.in/~sury/aisiit.pdf

[28] K. A. MIHAILOVA, *The occurrence problem for direct products of groups*. Dokl. Akad. Nauk SSSR (1958), pp.1103-1105.

[29] NEAL R. WAGNER AND MARIANNE R. MAGYARIK, *A public-key cryptosystem based on the word problem*. Advances in Cryptology, Proceeding of Cryto '84, Lecture Notes in Computer Science 196, ed. G. R. Blakley y D. Chaum, Springer-Verlag (1985), pp.19-36.
Disponible en web: https://link.springer.com/chapter/10.1007/3-540-39568-7_3